AllFusion™ Endevor® Change Manager

Parallel Development Option Guide

4.0

Computer Associates™

ENPDM400

First Edition, December 2002

Contents

Chapter 1. An Introduction to PDM 1-1
1.1 Overview 1-2
1.2 What is Parallel Development? 1-3
1.2.1 Simple Parallel Development 1-3
1.2.1.1 Problem 1-4
1.2.2 Complex Parallel Development 1-4
1.2.2.1 Problem 1-4
1.2.3 Vendor Application Updates 1-4
1.2.3.1 Problem 1-5
1.2.4 Integrating Concurrently Developed Software 1-5
1.3 The Parallel Development Option 1-6
1.3.1 PDM Tools for Project Analysis 1-6
1.3.2 PDM Tools for Consolidation and Conflict Resolution 1-6
1.3.3 PDM Tools for Integrating Changes 1-6
1.4 How PDM Works 1-7
1.5 Basic PDM Terminology 1-9
1.5.1 PDM File Names 1-9
1.5.2 PDM Terms 1-9
1.5.3 WIP Member Selection Matrix 1-10
1.6 Accessing PDM 1-11
1.7 Specifying User Defaults 1-12
1.7.1 PRTMAT Output Defaults 1-13
1.7.2 Disable the Build WIP and Merge Performance Enhancement Option 1-13
1.7.3 When You Press Enter 1-14
1.8 Documentation Overview 1-15
1.9 Name Masking 1-16
1.9.1 Usage 1-16
1.10 Syntax Conventions 1-18
1.10.1 Sample Syntax Diagram 1-21
1.10.2 Syntax Diagram Explanation 1-21
1.10.3 General Coding Information 1-23
1.10.3.1 Valid Characters 1-23
1.10.3.2 Incompatible Commands and Clauses 1-24
1.10.3.3 Ending A Statement 1-24
1.10.3.4 SCL Parsing Information 1-24

Chapter 2. Building WIP Files 2-1
2.1 The Build WIP Process 2-2
2.2 Specifying a WIP Library and Build WIP Options 2-4

2.2.1 WIP Library and Build WIP Options Panel Fields 2-5
2.2.1.1 WIP Library Specification Panel Fields 2-5
2.2.1.2 WIP Build Options 2-5
2.2.2 Allocating a WIP Library 2-6
2.2.2.1 WIP Data Set Allocation Panel Fields 2-7
2.2.3 Specifying a WIP Member 2-8
2.3 Specifying the Root 2-9
2.3.1 Specifying a Root Library or Endevor Location 2-9
2.3.2 Root Specification Panel Fields 2-9
2.3.3 Specifying a Root Member or Element 2-11
2.3.3.1 Selecting a Member 2-11
2.3.3.2 Selecting an Element 2-12
2.4 Specifying Derivation 1 2-14
2.4.1 Specifying a Derivation 1 Library or Endevor Location 2-14
2.4.2 Specifying a Derivation 1 Member or Element 2-14
2.4.2.1 Selecting Members or Elements 2-15
2.4.3 When You Exit the Derivation 1 Specification Panel 2-15
2.5 Specifying Derivation 2 2-17
2.5.1 Specifying a Derivation 2 Library or Endevor Location 2-17
2.5.2 Specifying a Derivation 2 Member or Element 2-17
2.5.2.1 Selecting Members or Elements 2-18
2.5.3 When You Exit the Derivation 2 Specification Panel 2-18
2.6 Creating a Merge Output File During the Build WIP Process 2-19
2.6.1 Merge Output Specification Panel Fields 2-20
2.7 Using the WIP Member Selection Matrix 2-21
2.7.1 Sample Matrix 2-21
2.7.2 Excluding a WIP Member 2-23
2.7.3 Including Matrix Rows 2-24
2.7.4 Renaming a WIP Member 2-25
2.7.5 Moving a Derivation 1 or Derivation 2 Member 2-26
2.7.6 Like-Named WIP Members 2-27
2.7.7 Printing the Matrix 2-29
Example 2-29
2.7.8 Positioning the Matrix 2-30
Example 2-30
2.7.9 Sorting the Matrix 2-31
Example 2-31
2.7.10 Creating WIP Files or SCL 2-31
2.8 Using the Summary of Levels Panel 2-33
2.8.1 Using the Summary of Levels Panel with the WIP Member Selection Matrix 2-33
2.8.1.1 Clearing an Assigned Version/Level 2-34
2.9 The Next Step 2-36

Chapter 3. Editing WIP Files 3-1
3.1 The Format of the WIP File 3-2
3.1.1 The WIP File Header 3-2
3.1.1.1 WIP File Header Field Descriptions 3-2
3.1.2 The WIP File 3-3
WIP File Annotations 3-3
3.2 Editing the WIP File 3-5

3.3 WIP Edit Commands . . . 3-7
3.3.1.1 WIP Edit Command Syntax . . . 3-7
3.3.1.2 WIP Edit Parameters . . . 3-7
3.3.1.3 WIP Edit Examples . . . 3-8
3.3.2 WIPCOUNT . . . 3-9
Purpose . . . 3-9
Execution . . . 3-9
Usage . . . 3-9
3.3.3 WIPLDEL . . . 3-9
Purpose . . . 3-9
Execution . . . 3-9
Usage . . . 3-9
3.3.4 WIPUNDEL . . . 3-9
Purpose . . . 3-9
Execution . . . 3-9
Usage . . . 3-9
3.3.5 WIPSHOW . . . 3-10
Purpose . . . 3-10
Execution . . . 3-10
3.3.6 WIPCHANG . . . 3-11
Purpose . . . 3-11
Execution . . . 3-11
Usage . . . 3-11
3.3.7 WIPPARA . . . 3-11
Purpose . . . 3-11
Execution . . . 3-11
Usage . . . 3-12
3.3.8 WIPCON . . . 3-12
Purpose . . . 3-12
Execution . . . 3-12
Usage . . . 3-12
3.3.9 WIPMERGE . . . 3-12
Purpose . . . 3-12
Execution . . . 3-13
Usage . . . 3-13
3.4 The Next Step . . . 3-14

Chapter 4. Merging WIP Files . . . 4-1
4.1 The Merge Output File . . . 4-2
4.1.1 WIP File . . . 4-3
4.1.2 Merge Output File Based on Edits to WIP File . . . 4-4
4.2 Creating a Merge Output File . . . 4-5
4.2.1 Procedure: Merging a WIP File . . . 4-5
4.3 The Merge Output Specification Panel . . . 4-8
4.3.1 Merge Output Specification Panel Fields . . . 4-8
4.3.1.1 The Merge Output Member Selection List . . . 4-9
4.3.1.2 Overlay Masks and the Selection List . . . 4-11
4.4 The Next Step . . . 4-12

Chapter 5. Using PDM in Batch . . . 5-1

5.1 Generating Build WIP and Merge Requests 5-2
5.2 Specifying a Request Data Set 5-3
5.2.1 Batch Request Data Set Panel Field Descriptions 5-3
5.2.2 Request Dataset Allocation Panel 5-4
5.2.2.1 Request Dataset Allocation Panel Field Descriptions 5-4
5.3 Generating Build WIP Requests Option 1 5-6
5.4 Generating Merge Requests Option 3 5-7
5.5 Submitting Batch Requests for Execution Option 4 5-8
5.5.1 The PDM Batch Options Menu 5-8
5.5.1.1 Browsing the Request Data Set 5-9
5.5.1.2 Editing the Request Data Set 5-10
5.5.1.3 Building JCL to be Included with the Batch Request 5-11
5.5.1.4 Submitting the Request Data Set for Batch Processing 5-11

Chapter 6. PDM Reports 6-1
6.1 General Information 6-2
6.2 PDM Syntax Request Report for Build WIP 6-3
6.2.1 Syntax Validation 6-3
6.3 PDM Build WIP Detail Report 6-4
6.3.1 Overview 6-4
6.3.2 PDM Build WIP Detail Report Fields Description 6-5
6.3.2.1 Beginning Build of WIP Information 6-5
6.3.2.2 Return Code Information 6-5
6.3.2.3 Detail Statistics 6-5
Relative Change 6-5
Inserts and Deletes 6-6
Conflicts 6-7
Build WIP Summary 6-7
6.4 PDM Build WIP Summary Report 6-9
6.4.1 Overview 6-9
6.4.2 PDM Build WIP Summary Report Field Descriptions 6-9
6.4.2.1 WIP Data Set Library Information 6-9
WIP Data Set Member Information 6-10
BUILD WIP Summary Information 6-10
6.4.3 Highest BUILD Return Code 6-11
6.5 PDM Syntax Request Report for Merge 6-13
6.5.1 Overview 6-13
6.6 PDM Merge Output Detail Report 6-14
6.6.1 Overview 6-14
6.6.2 PDM Merge Output Detail Report Field Descriptions 6-15
6.6.2.1 Beginning Merge of WIP Member Information 6-15
6.6.2.2 Return Code Information 6-15
6.6.2.3 Detail Statistics 6-15
6.7 PDM Merge Output Summary Report 6-16
6.7.1 Overview 6-16
6.7.2 PDM Merge Output Summary Report Field Descriptions 6-16
6.7.2.1 WIP Data Set Library Information 6-16
6.7.2.2 Output Data Set Information 6-16
6.7.2.3 WIP Member Information 6-16
6.7.2.4 Merge Summary Information 6-17
6.7.3 Highest MERGE Return Code 6-17

Appendix A. Batch Execution JCL A-1
A.1 Sample Batch Execution JCL A-2
A.1.1 Before Executing JCL A-2
A.1.2 Sample A-3

Appendix B. Batch Syntax B-1
B.1 PDM Syntax B-2
B.2 PDM Syntax Statements B-3
B.2.1 The Structure of a PDM Request Statement B-3
B.2.2 Process Flow B-4
B.3 The BUIld WIP Statement B-5
B.3.1 Parameters B-5
B.4 The BUIld MASK Statement B-10
B.4.1 Parameters B-10
B.5 The MERGE Statement B-12
B.5.1 Parameters B-12
B.6 The MERGE MASK Statement B-15
B.6.1 Parameters B-15
B.7 The STATS OFF Statement B-16
B.7.1 Parameters B-16
B.8 Syntax Examples B-17
EXAMPLE 1 . . . . . B-17
EXAMPLE 2 . . . . . B-18
EXAMPLE 3 . . . . . B-19

Appendix C. Extending the WIP Edit Macro Facility C-1
C.1 Before You Begin C-2
C.2 Existing Edit Macros C-3
C.3 Extending the Edit Macros C-4
C.3.1 BC1GM100 and BC1GMU01 . . . . . C-4

Appendix D. Statistical Data Control Blocks D-1
D.1 General Information D-2
D.2 $PBWSTAT: WIP Detail Statistics D-3
D.2.1 Data Displayed D-5
D.2.2 General Source Location Information D-6
D.2.3 General Statistics Total D-8
D.2.4 Data Common to Both Derivations D-9
D.2.5 Derivation 1 Statistics D-10
D.2.6 Derivation 2 Statistics D-11
D.2.7 Auto-Merge information D-11
D.3 $PBWSSUM: WIP Summary Statistics D-12
D.4 $PBMSTAT: Merge Detail Statistics D-14
D.4.1 Data Display D-15
D.4.2 General Source Location Information D-15
D.4.3 General Statistics D-16
D.5 $PBMSSUM: Merge Summary Statistics D-17

Index X-1

Chapter 1. An Introduction to PDM

1.1 Overview

The *Parallel Development Option Guide* explains how to use PDM to manage concurrent development and vendor application maintenance activities. The following sections are included in this chapter:

- What is Parallel Development?
- The Parallel Development Option
- How PDM Works
- Basic PDM Terminology
- Accessing PDM
- Specifying User Defaults

Also included in this chapter is general information regarding:

- Documentation Overview
- Name Masking
- Syntax Conventions

1.2 What is Parallel Development?

Parallel development is a term used to describe the concurrent development activities necessitated by the complexity of today's applications and the pace at which these applications are developed. Aspects of parallel development include:

- Several programmers working on the same application and often the same programs.
- Identifying and resolving conflicts in programs that have been independently updated by more than one programmer.
- The time-consuming, resource-intensive, and error-prone process of manually integrating independent changes to a program.
- Reconciling differences between customized vendor software packages and vendor release updates.
- Assessing project complexity, duration, and the resource requirements.

There are three general kinds of parallel development:

Term	Description
Simple parallel development	Concurrent work by a small number of programmers on modifications to a single program.
Complex parallel development	Concurrent work on multiple or overlapping releases by several teams of programmers.
Vendor application updates	Integrating customizations to a base release of vendor or internally developed software with a subsequent release of the software.

Each of these is discussed in the following sections. The discussion includes a problem typical of that kind of parallel development.

1.2.1 Simple Parallel Development

Simple parallel development is the concurrent and independent development of applications by different programmers within the same company. In most simple parallel development situations, modifications are minimal and conflicts are few or nonexistent. Managerial intervention is rarely required. Most of the time the developers themselves can determine the time needed to make the necessary changes.

1.2.1.1 Problem

Two programmers have been independently enhancing a single report generation program. Potential conflicts and overlaps need to be identified and resolved and the program modifications must be integrated before the program is placed into production.

1.2.2 Complex Parallel Development

Complex parallel development, like simple parallel development, involves more than one programmer working individually on the development of the same base program, with an eventual need to integrate the two sets of changes. But the actual complexity of the project--whether it be the degree of difficulty involved in the modifications required, the sheer number of changes involved, or the need for accurate and accessible information between all parties--sets it apart from a simple development situation. Complex parallel development can even involve two or more different sites. In this situation, the programs are not only being coded by different programmers, they are being coded by people who often have little or no communication with each other.

Complex parallel development occurs in situations such as these:

- A company must maintain an existing software release while developing a new release. For example, one team of developers must continue to write maintenance against an existing release, while another team of developers works on a new release of the product. Eventually, the maintenance changes must be incorporated in the next release.
- A company has multiple sites that are doing development at the same time. For example, a state university with campuses in several cities may be upgrading its admissions tracking program. Two of the campuses are making modifications that eventually need to be integrated into the existing admissions tracking program.

1.2.2.1 Problem

Modified copies of a program must be integrated to produce a single, updated package. In complex parallel development situations, the potential for conflict tends to be higher, and the changes more difficult. Management intervention is usually required to determine the best approach and the most appropriate resources to produce the final product.

1.2.3 Vendor Application Updates

Vendor products, such as proprietary or in-house developed products, need to be maintained and updated on a regular basis. Whether you have purchased a vendor package that was customized by the vendor before delivery, or customized the package yourself, you need to integrate periodic vendor updates into the modified vendor code.

1.2.3.1 Problem

The user customizations to the base product must be identified and carried forward into the new release.

1.2.4 Integrating Concurrently Developed Software

Integrating concurrently developed software requires varying levels of involvement by programmers and managers, depending on the complexity of the parallel development effort.

There are three steps to the integration process:

- Analysis, to determine the scope of the integration project. This can be handled by programmers in simple parallel development situations. More managerial involvement is often required for vendor updates or more complicated development projects.
- Consolidation and conflict resolution. Procedures need to be established for gathering the proper programs and identifying conflicts for resolution. Again, managerial involvement increases as project complexity increases.
- Integrating the changes. This is often a manual process, one that can be very time consuming and error prone in large or complicated projects.

1.3 The Parallel Development Option

The Parallel Development Option (PDM) is a powerful tool to help both developers and managers address the problems inherent in the three steps of parallel development projects. PDM can help:

- Analyze the complexity of parallel development projects.
- Consolidate changes and identify conflicts more quickly.
- Automate the integration process.

1.3.1 PDM Tools for Project Analysis

PDM produces a set of reports that managers can use for project planning. The reports are based on one or more Work-in-Process (WIP) files. PDM builds WIP files from a base file and one or two files derived from the base. The Build WIP process analyzes the input files, identifies differences and conflicts, and creates statistics and reports.

The PDM reports include information such as numbers of inserted and deleted lines in specific input files, and an assessment of the complexity of the WIP file. By interpreting the Build WIP reports, programmers or managers can more accurately plan the resources and time needed to complete the assignment.

1.3.2 PDM Tools for Consolidation and Conflict Resolution

The Build WIP process is the PDM mechanism for consolidating changes and identifying conflicts. By automating the consolidation of changes and identification of conflicts, PDM both speeds up and increases the accuracy of this step in parallel development.

PDM also provides editing tools to speed up the process of resolving any conflicts that exist.

1.3.3 PDM Tools for Integrating Changes

Once a WIP file has been edited to resolve conflicts, PDM can create an integrated output file by inserting and deleting lines based on the annotations in the WIP file, then write the integrated output file to a user specified location.

1.4 How PDM Works

The Parallel Development Option operates with standard PDS, PDSE and sequential data sets, as well as AllFusion CA-Panvalet files, AllFusion CA-Librarian files, and Endevor elements.

You can use PDM interactively (in foreground) or in batch. In either mode, PDM processing involves three basic steps:

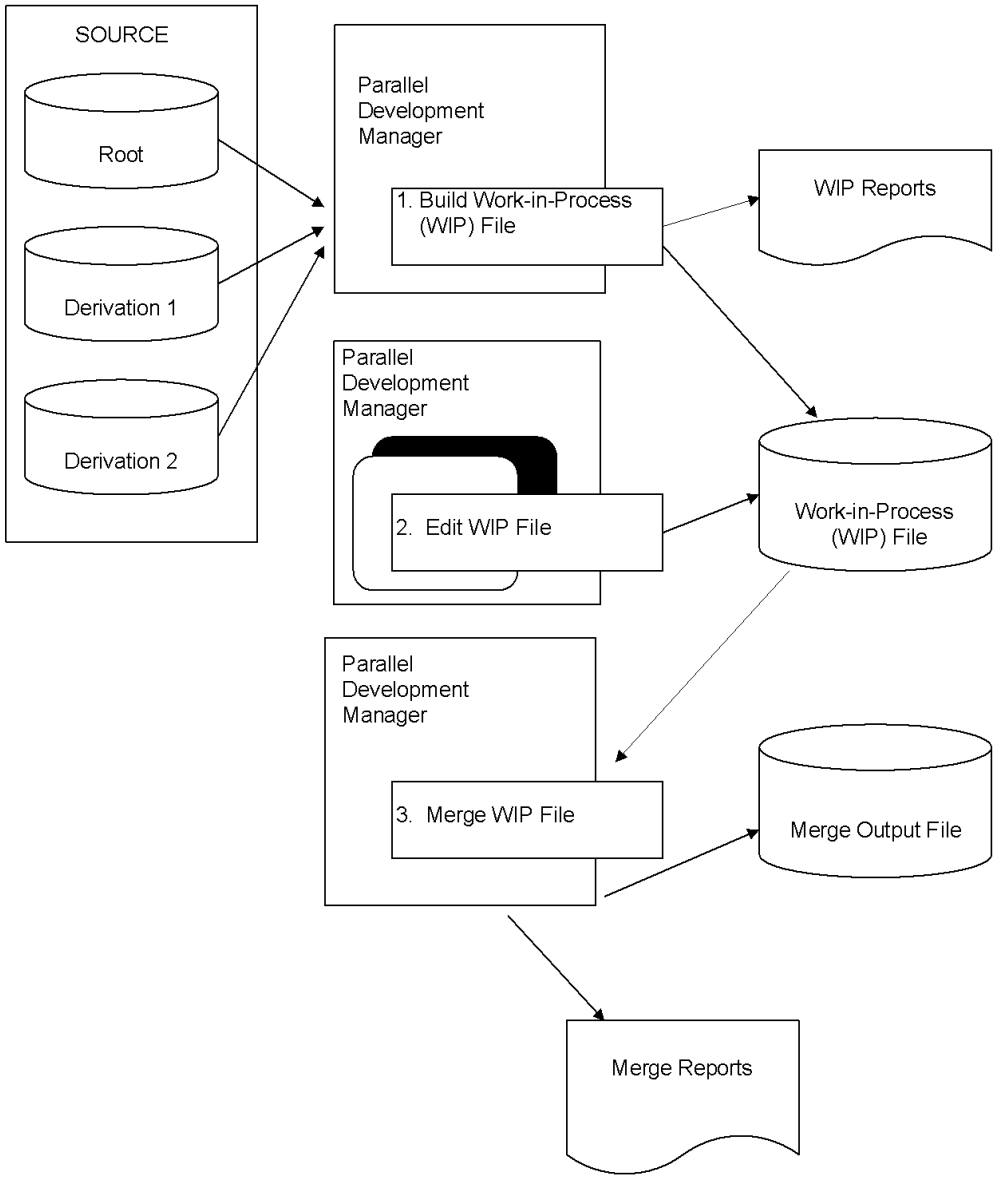
1. Build the Work-in-Process (WIP) File. PDM builds a Work-In-Process (WIP) file by comparing a base program and one or two files derived from the base. PDM annotates the WIP File to clearly mark all insertions, deletions, and conflict areas.

   The Build WIP process also produces reports and statistical data which allow a manager to estimate the time and resources required to resolve the conflicts.

2. Edit the WIP File. Programmers edit the WIP File to resolve annotated conflicts. Editing tools provided by PDM allow a project team to try different edit scenarios before actually integrating the changes in the WIP file.

3. Merge the WIP File. The final step in the PDM process is to merge the WIP File into a Merge Output File. A Merge Output File is a source file that can be stored in the appropriate source repository, input to a compiler, or added into Endevor. PDM uses the WIP File as input, performs the insertions and deletions noted, and produces the integrated source file.

The diagram below illustrates basic PDM operation.



Step 1: The Work-in-Process (WIP) File is created and reports are produced.

Step 2: The WIP File is edited and conflicts are resolved.

Step 3: The WIP File is integrated into a Merge Output File.

1.5 Basic PDM Terminology

Several PDM terms, including file names and reports, have already been mentioned in this chapter. To help familiarize you with these terms before you read the rest of the manual, they are defined here.

1.5.1 PDM File Names

Here's a list of PDM file names you should know.

File Name	Description
Root	The base program with which programmers are working. Derivation 1 and Derivation 2 files are derived from this program. For example, the Root might be Release 1.0 of a vendor product; Derivation 1, your modifications to the product; and Derivation 2, the vendor's new release (1.1) of the product.
Derivation 1	A modified copy of the Root.
Derivation 2	Another modified copy of the Root.
Work-In-Process (WIP) File	The file built when PDM compares the Root and one or two Derivations. The WIP File is an annotated intermediate file that identifies the insertions and deletions made individually by the Derivations, as well as the insertions and deletions common to both. Potential insert conflicts are also identified.
Merge Output File	An output file created by PDM using the WIP File as input. When the Merge Output File is created, the annotations from the WIP File are removed, and all insertions and deletions are performed.
Statistics File	An optional file that can be created by the Build WIP or Merge function. This file contains statistics that describe the members involved in the Build WIP or Merge operation. These statistics include items such as the number of insertions made by Derivation 1, the number of insertions made by Derivation 2, and the like. This file can be used as input to an existing report package or a user written reporting system.

1.5.2 PDM Terms

Here's a list of PDM terms you should know:

Term	Description
Common Insertion Common Deletion	The result of both programmers inserting or deleting the same line at the same place in their respective copies of the Root program

Term	Description
Conflict	The result of both programmers inserting different lines at exactly the same place in their respective copies of the Root program. Conflicts are identified in the WIP File.
Conflict Area	A place in the WIP File where a conflict occurs. A conflict area involves at least two records--one from Derivation 1 and one from Derivation 2.
Contention Area	A place in the WIP File where conflicting changes exist, but are "offset;" that is, both Derivations have changed a section of code, but the changes do not begin at exactly the same place in each derivation. Because these sections of code overlap, however, they are considered conflicting in nature.
Complexity Factor **1**	A value assigned to the WIP file, to help identify the resources required to resolve conflicts. The complexity factor is a function of the number of: ■ Records in the WIP file. ■ Derivation 1 records in conflict. ■ Derivation 2 records in conflict. ■ Conflict areas and contention areas. The complexity factor ranges from 0-5: ■ 0--indicates that no changes occurred in the member. ■ 1--indicates that changes occurred but there are no conflict or contention areas. ■ 2-5--indicate that the member contains conflicts, where 2 signifies simple conflicts and 5 signifies complex conflicts.

Note: **1** - The complexity factor reflects the number of conflicts found in the WIP File, not the number of changes. The comparison procedure may identify many changes, but few or no conflicts, resulting in a low complexity factor. Conversely, you may find that although you have a small number of conflicts, each is relatively complex, resulting in a high complexity factor.

1.5.3 WIP Member Selection Matrix

A special PDM screen that shows side-by-side alignment, within the WIP data set, of the Root, Derivation 1, and Derivation 2 members. The matrix is used when dealing with multiple sets of Root, Derivation 1, and Derivation 2 members.

1.6 Accessing PDM

To use PDM, select the program from the ISPF/PDF Primary Option Panel by entering the appropriate code.

The AllFusion Endevor Change Manager Parallel Development Option menu is used to select each step of the PDM process. This section summarizes each option on the menu.

```
-------  AllFusion Endevor Change Manager Parallel Development Option  --------
 Option ===>

    0  DEFAULTS  - Specify user session parameters
    1  BUILD WIP - Build Work-in-Process (WIP) file
    2  EDIT WIP  - Edit Work-in-Process (WIP) file
    3  MERGE     - Merge Work-in-Process (WIP) file into OUTPUT source file
    4  SUBMIT    - Build job for batch submission
    T  TUTORIAL  - Tutorial on how to use the Parallel Development Option
    C  CHANGES   - New features for this release

Processing Mode:
   Foreground or Batch ===> F  (Foreground/Batch)

--------------------------- Batch Specification --------------------
Batch Request Data Set:               Batch Options:
   Project  ===>                         Append or Replace ===>     (A/R)
   Group    ===>
   Type     ===>
   Member   ===>
 Other partitioned or sequential data set:
   Data Set Name ===>

            (C) 1987,2002 Computer Associates International, Inc.
```

Use this panel to indicate the primary option processing you want:

Option	Used to
0	Establish defaults for work-area space allocations. These defaults are user ID-specific.
1	Build a WIP File or write Build WIP requests to a batch request data set.
2	Edit a WIP file.
3	Merge a WIP file into an output source library. You can use either foreground or batch mode to merge the WIP File.
4	Build a job that executes Build WIP or Merge requests in batch.
T	Invoke an online tutorial that describes PDM procedures and panel fields.
C	View a summary of the new features that have been added to PDM for this release.

1.7 Specifying User Defaults

The PDM User Defaults panel allows you to override defaults for space allocation of work areas and temporary data sets that are used in foreground. The defaults remain in effect until you change them, either in this particular session or in a subsequent work session.

Allocation information is initially taken from the settings in the C1DEFLTS Table. The PDM User Defaults panel allows you to override those default values for your user ID only.

To select this option, type 0 in the option field of the AllFusion Endevor Change Manager Parallel Development Option menu.

```
-------  AllFusion Endevor Change Manager Parallel Development Option  --------
 Option ===> 0
    0  DEFAULTS  - Specify user session parameters
    1  BUILD WIP - Build Work-in-Process (WIP) file
    2  EDIT WIP  - Edit Work-in-Process (WIP) file
    3  MERGE     - Merge Work-in-Process (WIP) file into OUTPUT source file
    4  SUBMIT    - Build job for batch submission
    T  TUTORIAL  - Tutorial on how to use the Parallel Development Option
    C  CHANGES   - New features for this release

Processing Mode:
   Foreground or Batch ===> F  (Foreground/Batch)

--------------------------- Batch Specification --------------------
Batch Request Data Set:               Batch Options:
   Project  ===>                         Append or Replace ===>     (A/R)
   Group    ===>
   Type     ===>
   Member   ===>
 Other partitioned or sequential data set:
   Data Set Name ===>

            (C) 1987,2002 Computer Associates International, Inc.
```

When you press ENTER, the PDM User Defaults screen appears.

```
 Defaults ---------------------- PDM User Defaults -------------------------
 Command ===>

 Work Dataset Allocation Information:
   Primary Quantity   ===> 5
   Secondary Quantity ===> 5
   Space Units        ===> CYL    (TRK/CYL/BLK)
   Unit Name          ===> SYSDA
   Volume Serial      ===>        (Blank for default)

 PRTMAT Output Defaults:
   SYSOUT Class ===> A
   Destination  ===>                     or   Writer Name  ===>

 Disable the BUILD WIP and
   MERGE performance enhancement ===> N  (Yes/No)
```

These fields define the default allocation for your work.

Field	Description
Primary Quantity	Number of units of space in the primary allocation.
Secondary Quantity	Number of units of space in each secondary allocation.
Space Units	Units in which space is allocated. Acceptable values are TRK (tracks), CYL (cylinders) or BLK (blocks).
Unit Name	Descriptive name of the disk device. You can specify any value appropriate in the UNIT= parameter of DD (JCL) statements at your site.
Volume Serial	Volume serial number of the specific device you want to use for your work areas. Leave blank to use the site default.

1.7.1 PRTMAT Output Defaults

These fields allow you to set PDM defaults for the PRTMAT output destination for WIP Member Selection matrix output processing. The value of these fields depends on the standards established for your site.

Field	Description
SYSOUT Class	Required alphanumeric character defining the SYSOUT output class designation. This field is initialized to A.
Destination	Optional 1- to 17-character output destination identifier. The format is node.user. If only one parameter is used, PDM assumes that it designates the user. If blank, PDM uses the destination value LOCAL. This field is equivalent to the DEST= JCL statement. For example, to set the default PRTMAT destination to the locally defined printer PRINTER1, specify PRINTER1 in the destination field.

1.7.2 Disable the Build WIP and Merge Performance Enhancement Option

If PDM is executing on an OS/390 system, it uses virtual storage as a work area during certain build and merge operations. If the Root file, Derivation files, or the WIP file is extremely large, PDM may exhaust all available virtual storage and the Build or Merge action will fail.

The disable the build WIP and merge enhancement field controls whether PDM uses external (disk) storage instead of virtual storage during builds and merge processing.

Specify Y (yes) if PDM is to use external storage for its work area. Specify N (no) if PDM is to use virtual storage. The default is N.

Specify a value of Y in this field if PDM issues error message PDM2001E and the corrective action for this message does not correct the error.

1.7.3 When You Press Enter

When you press ENTER, the AllFusion Endevor Change Manager Parallel Development Option menu appears, with the following message in the upper right-hand corner of the screen:

```
Defaults Updated
```

This message indicates that the defaults have been updated.

1.8 Documentation Overview

This manual is part of a comprehensive documentation set that fully describes the features and functions of Endevor and explains how to perform everyday tasks. For a complete list of Endevor manuals, see the PDF Table of Contents file in the PDF directory, or the Bookmanager Bookshelf file in the Books directory.

The following section describes product conventions.

1.9 Name Masking

A name mask allows you to specify all names, or all names beginning with a particular string, to be considered when performing an action.

Name masks are valid on:

- Element names
- Report syntax
- ISPF panels
- API requests

Name masks are not valid on:

- Endevor location information (environment, system, subsystem, type and stage)
- Element names in the following situations:
  - When entering a LEVel in a statement
  - When using the MEMber clause with a particular action
  - When building a package

1.9.1 Usage

There are three ways to mask names: by using the wildcard character (*), by using the placeholder character (%), and by using both together.

The wildcard (*) can be used in one of two ways to specify external file names:

- When coded as the only character of a search string, Endevor returns all members of the search field. For example, if you coded the statement ADD ELEMENT *, all elements would be added.
- When coded as the last character of a search string, Endevor returns all members of the search field beginning with the characters in the search string preceding the wildcard. For example, the statement ADD ELEMENT UPD* would add all elements beginning with "UPD", such as UPDATED or UPDATE.

Note: You cannot use more than one wildcard in a string. The statement ADD ELEMENT U*PD* would result in an error.

The placeholder (%) can also be used in one of two ways:

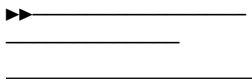
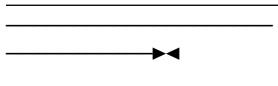
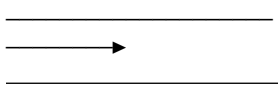
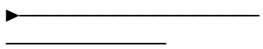
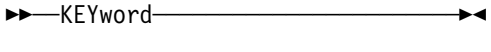
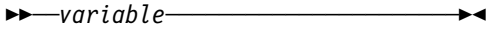
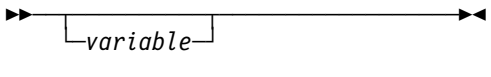
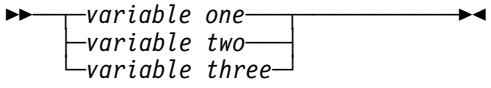
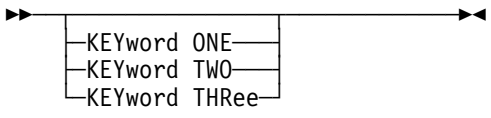
- When coded as the last character in a string, Endevor returns all members of the search field, beginning with the characters in the search string preceding the placeholder, but which have no more characters than were coded in the search string. If you coded the statement ADD ELEMENT UPD%, only those elements with four-character-long names beginning with "UPD" (UPD1 or UPDA, for example) would be added.

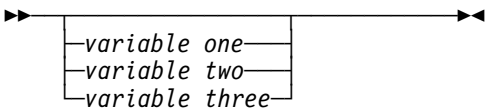
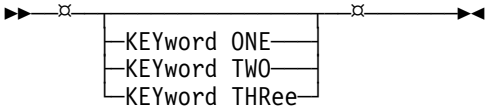
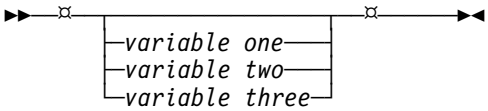
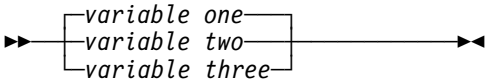
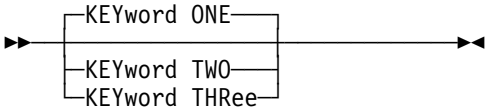
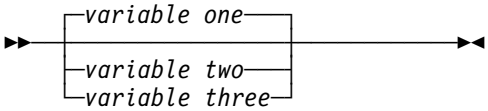
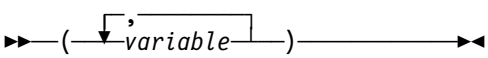
- It is also possible to use the placeholder multiple times in a single search string. The statement ADD ELEMENT U%PD% would return all elements with five-character-long names that have U as the first character, and PD third and fourth.

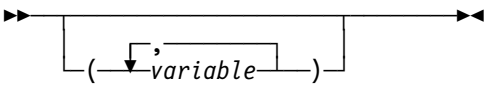
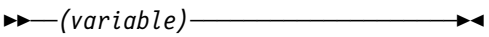
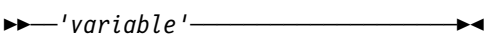
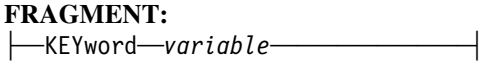
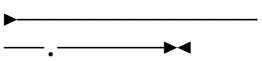
The wildcard and the placeholder can be used together, provided that the wildcard appears only at the end of the search string and is used only once. An example of a statement using both the wildcard and the placeholder is ADD ELEMENT U%D*. This statement would add elements with names of any length that have U as the first character and D as the third.

1.10 Syntax Conventions

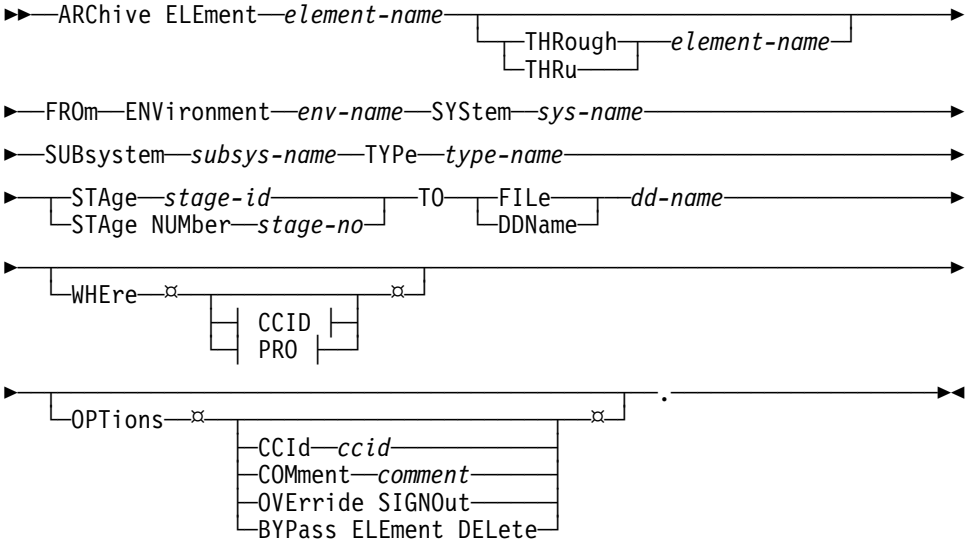
Endevor uses the IBM standard for representing syntax. The following table explains the syntax conventions:

Syntax	Explanation
	Represents the beginning of a syntax statement.
	Represents the end of a syntax statement.
	Represents the continuation of a syntax statement to the following line.
	Represents the continuation of a syntax statement from the preceding line.
	Represents a required keyword. Only the uppercase letters are necessary.
	Represents a required user-defined variable.
	Represents an optional keyword. Optional keywords appear below the syntax line. If coded, only the uppercase letters are necessary.
	Represents an optional user-defined variable. Optional variables appear below the syntax line.
	Represents a choice of required, mutually exclusive keywords. You must choose one and only one keyword.
	Represents a choice of required, mutually exclusive, user-defined variables. You must choose one and only one variable.
	Represents a choice of optional, mutually exclusive keywords. Optional keywords appear below the syntax line.

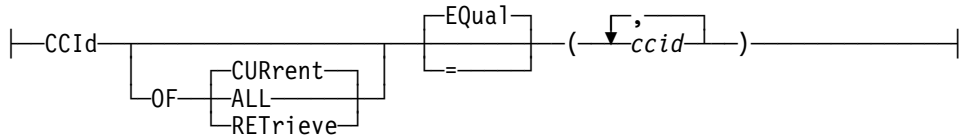
Syntax	Explanation
	Represents a choice of optional, mutually exclusive, user-defined variables. Optional variables appear below the syntax line.
	Represents a choice of optional keywords. The stars (¤) indicate that the keywords are not mutually exclusive. Only code each keyword once.
	Represents a choice of optional user-defined variables. The stars (¤) indicate that the variables are not mutually exclusive. Only code each variable once.
	Represents a choice of required, mutually exclusive keywords, one of which is the default. In this example, KEYword ONE is the default keyword because it appears above the syntax line.
	Represents a choice of required, mutually exclusive, user-defined variables, one of which is the default. In this example, *variable one* is the default variable because it appears above the syntax line.
	Represents a choice of optional, mutually exclusive keywords, one of which is the default. In this example, KEYword ONE is the default keyword because it appears above the syntax line.
	Represents a choice of optional, mutually exclusive, user-defined variables, one of which is the default. In this example, *variable one* is the default variable because it appears above the syntax line.
	Represents a required variable that can be repeated. Separate each occurrence with a comma and enclose any and all variables in a single set of parenthesis.

Syntax	Explanation
	Represents an optional variable that can be repeated. Separate each occurrence with a comma and enclose any and all variables in a single set of parenthesis.
	Represents a variable which must be enclosed by parenthesis.
	Represents a variable which must be enclosed by single quotes.
	Represents a variable which must be enclosed by double quotes.
	Represents a reference to a syntax fragment. Fragments are listed on the lines immediately following the required period at the end of each syntax statement.
FRAGMENT: 	Represents a syntax fragment.
	Represents the period required at the end of all syntax statements.

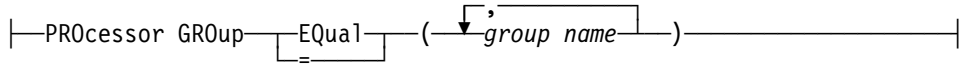
1.10.1 Sample Syntax Diagram



CCID:



PRO:



1.10.2 Syntax Diagram Explanation

Syntax	Explanation
ARChive ELEment *element-name*	The keyword ARChive ELEment appears on the main line, indicating that it is required. The variable *element-name*, also on the main line, must be coded.
THRough / THRu *element-name*	The keywords THRough and THRu appear below the main line, indicating that they are optional. They are also mutually exclusive.
FROm ENVironment ... TYPe *type-name*	Each keyword and variable in this segment appear on the main line, indicating that they are required.
STAge *stage-id* / STAge NUMber *stage-no*	The keywords STAge and STAge NUMber appear on and below the main line, indicating that they are required, mutually exclusive keywords.

Syntax	Explanation
TO ... *dd-name*	The keyword TO appears on the main line, indicating that it is required. The keywords FILe and DDName appear on and below the main line, indicating that they are required, mutually exclusive keywords. The variable *dd-name* also appears on the main line, indicating that it is required.
WHEre clause	This clause appears below the main line, indicating that it is optional. The keyword WHEre appears on the main line of the clause, indicating that it is required. CCID and PRO are syntax fragments that appear below the main line, indicating that they are optional. The stars (¤) indicate that they are not mutually exclusive. For details on the CCID and PRO fragments, see the bottom of this table.
OPTion clause	This clause appears below the main line, indicating that it is optional. The keyword OPTion appears on the main line of the clause, indicating that it is required. The keywords CCId, COMment, OVErride SIGNOut, and BYPass ELEment DELete all appear below the main line, indicating that they are optional. The stars (¤) indicate that they are not mutually exclusive.
CCID fragment	The keyword CCId appears on the main line, indicating that it is required. The OF clause appears below the main line, indicating that it is optional. If you code this clause, you must code the keyword OF, as it appears on the main line of the clause. CURrent, ALL, and RETrieve appear above, on, and below the main line of the clause, indicating that they are required, mutually exclusive keywords. CURrent appears above the main line, indicating that it is the default. If you code the keyword OF, you must choose one and only one of the keywords. The keywords EQual and = appear above and below the main line, indicating that they are optional, mutually exclusive keywords. EQual appears above the main line, indicating that it is the default. You can include only one. The variable *ccid* appears on the main line, indicating that it is required. The arrow indicates that you can repeat this variable, separating each instance with a comma. Enclose any and all variables in a single set of parenthesis.

Syntax	Explanation
PRO fragment	The keyword PROcessor GROup appears on the main line, indicating that it is required. The keywords EQual and = appear on and below the main line, indicating that they are required, mutually exclusive keywords. You must include one. The variable *group name* appears on the main line, indicating that it is required. The arrow indicates that you can repeat this variable, separating each instance with a comma. Enclose any and all variables in a single set of parenthesis.

1.10.3 General Coding Information

In coding syntax, you must adhere to certain rules and guidelines regarding valid characters, incompatible commands and clauses, and ending statements. In addition, knowing how the SCL parser processes syntax helps you resolve errors and undesired results. The following sections outline these rules and guidelines.

1.10.3.1 Valid Characters

The following characters are allowed when coding syntax:

- Uppercase letters
- Lowercase letters
- Numbers
- National characters
- Hyphen (-)
- Underscore (_)

The following characters are allowed when coding syntax, but must be enclosed in either single (') or double (") quotation marks:

- Space
- Tab
- New line
- Carriage return
- Comma (,)
- Period (.)
- Equal sign (=)
- Greater than sign (>)
- Less then sign (<)

- Parenthesis ( )
- Single quotation marks
- Double quotation marks

A string containing single quotation marks must be enclosed in double quotation marks. A string containing double quotation marks must be enclosed in single quotation marks.

To remove information from an existing field in the database, enclose a blank space in single or double quotation marks. For example, the following statement removes the default CCID for user TCS:

```
DEFINE USER TCS
DEFAULT CCID " ".
```

The characters "*" and "%" are reserved for name masking. See 1.9, “Name Masking” on page 1-16 for more information.

1.10.3.2 Incompatible Commands and Clauses

The following commands and clauses are mutually exclusive:

- THRough and MEMber clauses within any action except LIST
- Endevor location information (environment, system, subsystem, type, and stage) and data set names (DSName)
- File names (DDName) and data set names (DSName)
- The stage id (STAge / STAge ID) and the stage number (STAge NUMber)
- The SET TO Endevor location information and the SET TO MEMber clause

1.10.3.3 Ending A Statement

You must enter a period at the end of each statement. If no period is found, you receive an error message and the job terminates.

1.10.3.4 SCL Parsing Information

- The SCL parser does not look for information in columns 73-80 of the input. Therefore, be sure that all relevant information is coded in columns 1-72.
- The SCL parser does not catch duplicate clauses coded for an SCL request. If you code the same clause twice, SCL uses the Boolean "AND" to combine the clauses. If the result is invalid, you receive an error message.
- If you enter an asterisk (*) in column 1, the remainder of the line is considered a comment by the SCL parser and is ignored during processing.
- Any value found to the right of the period terminating the SCL statement is considered a comment by the SCL parser and is ignored during processing.

Chapter 2. Building WIP Files

2.1 The Build WIP Process

The first step in using the Parallel Development Option is to build the Work-in-Process (WIP) file and analyze the results. The WIP file is produced by comparing the Root (or base) file against one or two Derivation (modified) files. The WIP file combines the Root file and every insertion and deletion that has been made to the Derivation file(s), noting conflicting changes.

When building the WIP file, PDM uses the Derivation 1 and Derivation 2 files to determine which lines to include in the WIP file. This means that

If there is a	PDM Writes the Line From
Common insertion (% I-1,2)	Derivation 2
Insertion from Derivation 2 (% I-2)	Derivation 2
Insertion from Derivation 1 (% I-1)	Derivation 1
No change from the Root	Derivation 2 (or Derivation 1 if there is a two-way compare.)

The Build WIP process produces one or more of the following reports:

- PDM Syntax Report
- PDM Build WIP Detail Report
- PDM Build WIP Summary Report

These reports are discussed in Chapter 6, “PDM Reports.”

Note: PDM can produce a Build WIP Summary Report during the Build WIP process without actually creating the WIP file. Refer to the discussion on WIP Specification later in this chapter.

To begin the Build WIP process, select option **1** from the AllFusion Endevor Change Manager Parallel Development Option menu.

```
-------  AllFusion Endevor Change Manager Parallel Development Option  --------
 Option ===>

    0  DEFAULTS  - Specify user session parameters
    1  BUILD WIP - Build Work-in-Process (WIP) file
    2  EDIT WIP  - Edit Work-in-Process (WIP) file
    3  MERGE     - Merge Work-in-Process (WIP) file into OUTPUT source file
    4  SUBMIT    - Build job for batch submission
    T  TUTORIAL  - Tutorial on how to use the Parallel Development Option
    C  CHANGES   - New features for this release

Processing Mode:
   Foreground or Batch ===> F  (Foreground/Batch)

---------------------------- Batch Specification --------------------
Batch Request Data Set:               Batch Options:
   Project  ===>                         Append or Replace ===>     (A/R)
   Group    ===>
   Type     ===>
   Member   ===>
 Other partitioned or sequential data set:
   Data Set Name ===>

            (C) 1987,2002 Computer Associates International, Inc.
```

When you select option 1, you must also indicate in the FOREGROUND OR BATCH field whether you want to create the WIP File in foreground or batch mode. Enter **F** to indicate foreground; enter **B** to indicate batch. If you specify **B**, you must also provide request data set information.

See Chapter 5, “Using PDM in Batch” for information about the batch specification fields.

There are up to five steps in the Build WIP process:

Step	Action
1	Specify a WIP library and build WIP options.
2	Specify a Root library.
3	Specify a Derivation 1 library.
4	Optionally, specify a Derivation 2 library.
5	Optionally, specify a Merge Output data set.

These steps are discussed in the following sections.

2.2 Specifying a WIP Library and Build WIP Options

When you type **1** in the OPTION field and press ENTER, the WIP Specification panel displays.

```
Build WIP ----------------- WIP Specification Panel ------------------------
 Command ===>

 Specify Work-in-Process ("WIP") library below, then press the ENTER key.

 WIP Library:
    Project  ===> ENDEVOR
    Group    ===> DEV
    Type     ===> WIPLIB
    Member   ===>

 WIP Other partitioned or sequential data set:
    Data Set Name  ===>

 Data set options:
    If partitioned, replace like-named members  ===> Y     (Yes/No)

 WIP Build options:
    Number of Derivation files ===> 2         (1/2)
    Create WIP                 ===> Y         (Yes/No)
    Automatically Merge        ===> N         (Yes/No)
    Generate WIP reports       ===> A         (All/Summary/No)
    Write stats to data set ===>
```

The WIP Specification Panel identifies the library or data set containing the WIP files you build.

2.2.1 WIP Library and Build WIP Options Panel Fields

2.2.1.1 WIP Library Specification Panel Fields

Field	Description
WIP Library	Required. Using standard ISPF naming conventions, specifies the library containing the new WIP files. Optionally, you can enter MEMBER information; this is explained later in this section. The WIP data set must be a partitioned or sequential data set. Its LRECL is equal to the LRECL of the Root library plus eight bytes. **Example** The Root library's LRECL is 80 bytes, the WIP data set's LRECL is 88 (80 + 8).
WIP Other Partitioned or Sequential Data Set	This field is an alternative to entering WIP LIBRARY information, using standard ISPF conventions.
If Partitioned, Replace Like-Named Members	Indicates whether you want to replace like-named members in the WIP data set. Acceptable values are: ■ Y — Default. Replace like-named members. ■ N — Do not replace like-named members. This option does not apply to a sequential WIP file.

2.2.1.2 WIP Build Options

The WIP Build options fields allow you to select several options for WIP Build processing.

Field	Description
Number of Derivation Files	Required. Indicates the number of modified files the Root file is compared against. Valid values are: ■ 1 — One file. ■ 2 — Default. Two files.
Create WIP	Required. Indicates if PDM should write a WIP file. Value values are: ■ Y — Default. PDM creates and writes to the WIP file. ■ N — PDM bypasses writing to the WIP file, and still produces all the requested PDM reports and statistics.

Field	Description
Automatically Merge	Required. Indicates if PDM should automatically merge the WIP data set. Valid values are: ■ N — Default. Do not merge the WIP data set. ■ Y — Automatically merge the WIP data set if no conflicts are detected.
Generate WIP Reports	Required. Valid values are: ■ A (All) — Default. Generates both the WIP Summary and Detail Reports. ■ S (Summary) — Generates only the WIP Summary Report. ■ N (No) — Prevents the generation of any reports.
Write Stats to Data Set	Optional. Specifies the name of the data set used to capture PDM statistics. It must conform to the normal ISPF/PDF naming conventions. See Appendix D, “Statistical Data Control Blocks” for statistics data set characteristics.

2.2.2 Allocating a WIP Library

If the WIP library you specify on the WIP Specification panel is not cataloged, PDM displays the WIP Dataset Allocation panel. This panel allows you to create a new WIP data set with the characteristics that you specify on the panel.

```
Build WIP -------------- WIP Dataset Allocation Panel   Dataset not Catalogued
 Command ===>

 Dataset Name: ENDEVOR.PDM.WIPLIB
   Management Class     ===>             (Blank for default management class)
   Storage Class        ===>             (Blank for default storage class)
     Volume Serial      ===>             (Blank for authorized default volume)
   Data Class           ===>             (Blank for default data class)
     Space Units        ===>             (BLKS, TRKS, CYLS)
     Primary Quantity   ===>             (In above units)
     Secondary Quantity ===>             (In above units)
     Directory Blocks   ===>             (Zero for sequential data set)
     Record Format      ===> FB          (F, FB, V, VB)
     Record Length      ===> 88          *See note below
     Block Size         ===> 14520       (Blank for system determined value;
                                          Required when Space Units is BLKS)
   *Note: The WIP dataset record length must be at least 8 bytes larger than
    the largest of the Root, Derivation 1 and Derivation 2 record lengths.
    The maximum record length is 32000 (fixed length) or 32004 (variable length).
```

2.2.2.1 WIP Data Set Allocation Panel Fields

Field	Description
Management Class	Optional. If you specify a management class that is not defined, the allocation fails and PDM issues an error message.
Storage Class	Optional. If you specify a storage class that is not defined, the allocation fails and PDM issues an error message.
Volume Serial	Optional. If you specify a volume serial that is not defined, the allocation fails with a recoverable application error.
Data Class	Optional. If you specify a data class that is not defined, the allocation fails and PDM issues an error message.
Space Units	Acceptable values are: ■ BLKS — Blocks ■ TRKS — Tracks ■ CYLS — Cylinders
Primary Quantity	Numeric field. Space is allocated in the units specified in the SPACE UNITS field.
Secondary Quantity	Numeric field. Space is allocated in the units specified in the SPACE UNITS field.
Directory Blocks	Numeric field. ■ Non-zero values are required when specifying a WIP member ■ Zero — Required for a sequential data sets.
Record Format	Acceptable values are: ■ F ■ FB ■ V ■ VB
Record Length	Must be numeric, greater than zero, and less than 32000 (fixed) or 32004 (variable)
Block Size	Required if the space unit is BLKS (blocks), otherwise optional. If left blank, the block size is set to zero. If provided, the value must be numeric, and greater than zero. If the block size is specified and is not zero, and the record format is: ■ F — The block size must be equal to the record length. ■ FB — The block size must be an integral multiple of the record length. ■ V or VB — The block size must be at least four bytes larger than the record length.

2.2.3 Specifying a WIP Member

You have three alternatives when entering member names in conjunction with the WIP library. You can:

- **Leave the field blank**. PDM builds a WIP library member list that is identical to the Root library member list.
- **Type a specific member**. If you type a specific member name, the WIP File is built for that member only.
- **Type an overlay mask**. An overlay mask differs from a standard name mask in that the characters specified before the asterisk (*) *overlay* the corresponding characters in the Root member name.

  For example, if you provide an overlay mask of **W***, every WIP member name begins with the letter **W**. This means that Root member BC1PAL00 appears in the WIP library as member WC1PAL00.

Press ENTER when you have completed all the WIP specification data. The Root Specification Panel appears next.

2.3 Specifying the Root

When you press ENTER after specifying a WIP library and WIP build options, the Root Specification panel displays.

```
BUILD WIP ---------------- Root Specification Panel ------------------------
 Command ===>

 Specify Root location below, then press the ENTER key.

 Build Root from  ===> D     (Data set/Endevor)

 Root Library:                                          Compare Columns:
    Project  ===>                                          From      ===>
    Group    ===>                                          Through   ===>
    Type     ===>
    Member   ===>                 Thru Member ===>

 Root Other partitioned or sequential data set:
    Data Set Name ===>
 ENDEVOR:                             List Options:
    Environment  ===>                   Where CCID  ===>
    System       ===>                   CCID Type   ===> A  (Any/Base/Generate/
    Subsystem    ===>                                        Retrieve/Last action)
    Element      ===>                  Version  ===>    Level  ===>
    Type         ===>
    Stage        ===>
```

2.3.1 Specifying a Root Library or Endevor Location

Use the Root Specification Panel to define the Root library (and members) or the Root Endevor location (and elements) you want to use when building the WIP file.

Note: You can specify a Root library and member(s) or a Root Endevor location and element(s), but not both.

2.3.2 Root Specification Panel Fields

Enter the following information.

Field	Description
Build Root from	Indicates if the Root source is built from a data set or Endevor. Valid values are: ■ D — Default. The root source is built from a data set. ■ E — Endevor.
Root Library	Identifies the library containing the Root file(s). Define the library using standard ISPF naming conventions.
Root Other Partitioned or Sequential Data Set	This field can be used as an alternative to entering ROOT LIBRARY information.

Field	Description
Compare Columns	Required. Indicates the columns to be compared (from and through) during the Build WIP operation. It is very important that you specify the correct compare column values for the data you are processing. For example, columns 1-6 of a COBOL program usually contain sequence numbers. You should therefore specify a compare range of columns 7-72 for COBOL. PDM then builds the WIP file based on the contents of columns 7 through 72. Likewise, if the Root is an assembler program, the compare range should be columns 1-72. If the Root is an Endevor location, PDM uses the compare column information on the type definition for that element. In addition, if the compare through value specified on the element type definition is greater than the Root or Derivation data set length or the element source length, PDM writes an error message and terminates the Build WIP process. **Note:** Use caution when specifying a compare range for variable length records. If record lengths are different, PDM may annotate a record (line) as changed, even if record data is identical. For example, trailing blanks may be truncated in a variable length record.
Endevor fields	These fields must be explicit: ■ ENVIRONMENT ■ SYSTEM ■ SUBSYSTEM ■ TYPE Selection lists are provided if necessary. If you provide an invalid Stage ID, acceptable stage ID values are displayed to the right of this field after you press ENTER. See 2.3.3, “Specifying a Root Member or Element” on page 2-11 for instructions on providing these element names.
Version and Level	These fields allow you to build a WIP member using a specific version and level of an element. You can: ■ Type a version and level on the Root Specification Panel. ■ Use a Summary of Levels panel to select a version/level of the element. See the 2.8, “Using the Summary of Levels Panel” on page 2-33 for instructions.

Field	Description
Where CCID and CCID Type fields	These fields allow you to specify a CCID as a selection criterion when building a list of elements. The CCID TYPE field allows you to further qualify the CCID. The valid values are: ■ A — Default. Any matching CCID field. ■ B — The Base CCID field. ■ L — The Last Action CCID field. ■ G — The Generate CCID field. ■ R — The Retrieve CCID field. CCID selection is done only when a CCID is specified in the WHERE CCID field.

2.3.3 Specifying a Root Member or Element

Specify a Root member or element according to the following table.

If the WIP Library Member Was Specified	You Can Specify Root Members or Elements
By leaving the field blank, or By using an overlay mask.	By leaving the field blank, or Using a name mask, or As an explicit value.
As an explicit value.	Only as an explicit value.

Note: If the WIP library is a sequential data set, you must specify an explicit member or element name.

2.3.3.1 Selecting a Member

If you provide an explicit member name for the WIP library, but do not specify an explicit member at the Root location, PDM displays a Build WIP - Member Selection List.

```
BUILD WIP -----------  BUILD WIP - MEMBER SELECTION LIST  -  ROW 1 TO 11 OF 11
 COMMAND ===>                                               SCROLL ===> PAGE

  Select Member For BUILD WIP operation:

    ROOT  DSN: ENDEVOR.PDM.ROOT

     NAME                 VV.MM  CREATED     CHANGED       SIZE   INIT    ID
    JEM1                  01.01 01/02/06  02/03/06 10:51      5      8 NA1JM44
    JEM2                  01.00 01/02/06  02/01/06 10:58      7      7 NA1JM44
    JEM3                  01.00 01/02/06  02/02/06 11:02     22     22 NA1JM44
    WIPTST13              01.00 01/01/21  02/07/21 09:35      3      3 DA1MF45
    WIP46                 01.00 01/02/06  02/01/06 13:24      6      6 DA1MF45
    WIP47                 01.00 01/02/06  02/01/06 13:24      6      6 DA1MF45
    WIP48                 01.00 01/02/06  02/01/06 13:24      6      6 DA1MF45
    WIP49                 01.00 01/02/06  02/01/06 13:24      6      6 DA1MF45
    W1                    01.00 01/01/10  02/03/10 15:05      4      4 DA1MF45
    W2                    01.00 01/01/10  02/01/10 15:08      4      4 DA1MF45
    W3                    01.00 01/01/23  02/01/23 15:15    104    104 DA1MF45
 ****************************** BOTTOM OF DATA ******************************
```

PDM allows you to select only one member from this list. After you select a member, you can do one of the following:

- Press PF3 to return to the Root Specification Panel. Pressing PF3 deselects the member.
- Press ENTER to display the Derivation 1 Specification Panel.

2.3.3.2 Selecting an Element

If you provide an explicit member name for the WIP library, but do not specify an explicit element at the Root Endevor location, PDM displays an Element Selection List. The list contains the current level of the selected elements at the specified location.

```
Build WIP ----------------- Element Selection List --------  ROW 1 TO 14 OF 14
 Command ===>                                                 Scroll ===> PAGE

   Environment: SMPLTEST           System: ADMIN            Subsystem: PROCESS
                                   Type:   ISPPE            Stage: T

   Element    VV.LL   -Base  CCID-   -------- Current Level Comment ---------
   C1SDAG00   01.01   36-0949        ADDING AFTER CHANGING HELP TO CITRQST0
   C1SDGJ00   01.04   36-0949        ADDING AFTER CHANGING HELP TO CITDSEL0
   C1SDPR00   01.02   36-0949        ADDING AFTER CHANGING HELP TO CITRQST0
   C1SDSB00   01.03   36-0949        ADDING AFTER CHANGING HELP TO CITRQST0
   C1SDSY00   01.04   36-0949        ADDING AFTER CHANGING HELP TO CITRQST0
   C1SDTY00   01.03   36-0949        ADDING AFTER CHANGING HELP TO CITRQST0
   C1SD00NA   01.04   ZSXGMG1        FIX BAD ATTR LINE
   C1SD0000   01.06   ZSXGMG1        ADD 15 TUTORIALS
   C1SD10NA   01.11   ZSXGMG1        ADD 15 TUTORIALS
   C1SD1000   01.13   36-500         CHANGE SEARCH AND JUMP LITERALS
   C1SD2000   01.02   HELP           CHANGE HELP TUTORIAL LOGIC
   C1SD400C   01.02   ENVMAP         UPDATE FOOTPRINT DISPLAY PANELS
   C1SD4000   01.07   ENVMAP         UPDATE FOOTPRINT DISPLAY PANELS
   C1SD4100   01.03   ENVMAP         UPDATE FOOTPRINT DISPLAY PANELS
 ****************************** BOTTOM OF DATA ******************************
```

PDM allows you to select only one element from this list. To access a Summary of Levels panel for an element, first type **S** in the COMMAND field, then type **S** next to the desired element, and press ENTER.

Note: If the Root Specification Panel reappears with the message "Type "S" for Levels," and **S** is preloaded in the COMMAND field, press ENTER to access a Summary of Levels panel.

After you have selected an element you can do one of the following:

- Press PF3 to return to the Root Specification Panel. Pressing PF3 deselects the element.
- Press ENTER to display the Derivation 1 Specification Panel.

2.4 Specifying Derivation 1

The Derivation 1 Specification Panel appears when you press ENTER after completing the Root Specification Panel.

```
Build WIP ------------- Derivation 1 Specification Panel ----------------------
  Command ===>

  Specify Derivation 1 location below, then press the ENTER key.

  Build Derivation 1 from  ===> D     (Data set/Endevor)

  Derivation 1 Library:
     Project  ===>
     Group    ===>
     Type     ===>
     Member   ===>                    Thru Member ===>

  Derivation 1 Other partitioned or sequential data set:
     Data Set Name ===>

  ENDEVOR:                           List Options:
     Environment  ===>                 Where CCID  ===>
     System       ===>                 CCID Type   ===> A  (Any/Base/Generate
     Subsystem    ===>                                      Retrieve/Last action)
     Element      ===>                Version  ===>    Level  ===>
     Type         ===>
     Stage        ===>
```

2.4.1 Specifying a Derivation 1 Library or Endevor Location

Use the Derivation 1 Specification Panel to define the Derivation 1 source location (and members) you want to use when building the WIP File. The fields on this panel are described in the section Specifying a Root Library or Endevor Location, earlier in this chapter.

2.4.2 Specifying a Derivation 1 Member or Element

Specify a Derivation 1 library member according to the following table.

If the Root Library Member or Element is Specified	You Can Specify the Derivation 1 Member or Element
By one of these methods: Leaving the field blank Using an overlay mask	Using one of these methods: Leaving the field blank Using a name mask An explicit value
As an explicit value.	Only as an explicit value.

2.4.2.1 Selecting Members or Elements

If you provide an explicit member or element name for the Root location, but do not do so at the Derivation 1 location, PDM displays a member or element selection list. The use of these selection lists is described in 2.3.3.1, “Selecting a Member” on page 2-11 and 2.3.3.1, “Selecting a Member” on page 2-11.

2.4.3 When You Exit the Derivation 1 Specification Panel

When you have specified all Derivation 1 source information and press ENTER, one of the following occurs:

- The Derivation 2 Specification Panel appears if you specified NUMBER OF DERIVATION FILES=2 on the WIP Specification panel. See 2.5, “Specifying Derivation 2” for more information.
- The Merge Output Specification panel appears if you specified NUMBER OF DERIVATION FILES=1 and AUTOMATICALLY MERGE=Y on the WIP Specification panel.

  To proceed, specify a library to which PDM can write the merged output from the WIP file. See 2.6, “Creating a Merge Output File During the Build WIP Process” on page 2-19 for more information.
- The AllFusion Endevor Change Manager Parallel Development Option menu appears if you provided explicit Root library and Derivation 1 library member names, and specified NUMBER OF DERIVATION FILES=1 and AUTOMATICALLY MERGE=N on the WIP Specification panel.

  The AllFusion Endevor Change Manager Parallel Development Option menu displays the message WIP BUILD COMPLETE, meaning that PDM has created a WIP file from the Root and Derivation 1 library members.
- The message log appears if the Build WIP process fails, and you specified NUMBER OF DERIVATION FILES=1 and AUTOMATICALLY MERGE=Y on the WIP Specification panel.
- The AllFusion Endevor Change Manager Parallel Development Option menu appears if the Build WIP process fails, and you specified NUMBER OF DERIVATION FILES=1 and AUTOMATICALLY MERGE=Y on the WIP Specification panel, and there are conflicts in the WIP file.
- The WIP Member Selection Matrix appears if you specified:
  - NUMBER OF DERIVATION FILES=1 and AUTOMATICALLY MERGE=N on the WIP Specification panel
  - The WIP Library by leaving the MEMBER field blank or using an overlay mask
  - The Root library member by leaving the MEMBER field blank or using an overlay mask
  - The Derivation 1 library member by leaving the MEMBER field blank, using an overlay mask, or providing an explicit value

See 2.7, “Using the WIP Member Selection Matrix” on page 2-21 for more information.

2.5 Specifying Derivation 2

The Derivation 2 Specification Panel appears when you press ENTER after completing the Derivation 1 Specification panel and if you specified NUMBER OF DERIVATION FILES=2 on the WIP Specification panel.

```
Build WIP ------------- Derivation 2 Specification Panel -------------------
  Command ===>
  Specify Derivation 2 location below, then press the ENTER key.
  Build Derivation 2 from  ===> D     (Data set/Endevor)
  Derivation 2 Library:
     Project  ===>
     Group    ===>
     Type     ===>
     Member   ===>                   Through Member ===>
  Derivation 2 Other partitioned or sequential data set:
     Data Set Name ===>
  ENDEVOR:                         List Options:
     Environment  ===>               Where CCID  ===>
     System       ===>               CCID Type   ===> A (Any/Base/Generate
     Subsystem    ===>                                    Retrieve/Last action)
     Element      ===>               Version  ===>    Level  ===>
     Type         ===>
     Stage        ===>
```

2.5.1 Specifying a Derivation 2 Library or Endevor Location

Use the Derivation 2 Specification Panel to define the Derivation 2 library (and members) or the Derivation 2 Endevor location (and elements) you want to use when building the WIP File. The fields on this panel are described in 2.3.1, “Specifying a Root Library or Endevor Location” on page 2-9.

2.5.2 Specifying a Derivation 2 Member or Element

Specify a Derivation 2 library member according to the following table.

If the Root Member or Element Was Specified	You Can Specify the Derivation 1 Member or Element	And You Can Specify the Derivation 2 Member or Element
By one of these methods: Leaving the field blank Using a name mask	Using one of these methods: Leaving the field blank Using a name mask An explicit value	Using one of these methods: Leaving the field blank Using a name mask An explicit value
As an explicit value.	Only as an explicit value.	Only as an explicit value.

2.5.2.1 Selecting Members or Elements

If you provide an explicit member or element name for the Root location, but do not do so at the Derivation 1 location, PDM displays a member or element selection list. The use of these selection lists is described in the sections Selecting a Member and Selecting an Element earlier in this chapter.

2.5.3 When You Exit the Derivation 2 Specification Panel

When you have specified all Derivation 2 source information and press ENTER, one of the following happens:

- The Merge Output Specification panel appears if you specified NUMBER OF DERIVATION FILES=2 and AUTOMATICALLY MERGE=Y on the WIP Specification panel.

  To proceed, specify a library to which PDM can write the merged output from the WIP file. See 2.6, “Creating a Merge Output File During the Build WIP Process” on page 2-19 for more information.

- The AllFusion Endevor Change Manager Parallel Development Option menu appears if you provided explicit Root library, Derivation 1 and Derivation 2 library member names, and specified NUMBER OF DERIVATION FILES=2 and AUTOMATICALLY MERGE=N on the WIP Specification panel.

  The AllFusion Endevor Change Manager Parallel Development Option menu displays the message WIP BUILD COMPLETE, meaning that PDM has created a WIP file from the Root, Derivation 1 and Derivation 2 library members.

- The message log appears if the Build WIP process fails, and you specified NUMBER OF DERIVATION FILES=2, AUTOMATICALLY MERGE=N on the WIP Specification panel, and the WIP, Root, and Derivation 1 files were explicitly specified.

- The WIP Member Selection Matrix appears if you specified
  - NUMBER OF DERIVATION FILES=2 on the WIP Specification panel
  - The Root library member by leaving the MEMBER field blank or using an overlay mask
  - The Derivation 1 and Derivation 2 library members by leaving the MEMBER field blank, using an overlay mask, or providing an explicit value.

    See 2.7, “Using the WIP Member Selection Matrix” on page 2-21 for more information.

2.6 Creating a Merge Output File During the Build WIP Process

If you specify AUTOMATICALLY MERGE=Y on the WIP Specification panel, PDM will try to create automatically a merge file from the WIP file as the last step in the Build WIP process. PDM will create the Merge file only if it finds no conflicts in the WIP file.

When you specify AUTOMATICALLY=Y on the WIP Specification Panel, the Merge Output Specification Panel appears when you press ENTER after providing necessary information on either the Derivation 1 or Derivation 2 Specification panel.

```
Build WIP ------------ Merge Output Specification Panel --------------------
 Command ===>

 Specify MERGE OUTPUT location below, then press the ENTER key.

 Merge Library:
    Project ===> ENDEVOR
    Group   ===> PDM
    Type    ===> MERGELIB

 Merge Other partitioned or sequential data set:
    Data Set Name  ===>

 Merge Data Set Options:
    If partitioned, replace like-name members ===> Y    (Yes/No)

 Merge Options:
    Delete WIP member after MERGE ===> N    (Yes/No)
    Panvalet/Librarian language   ===>
```

Use the Merge Output Specification Panel to:

- Identify the library to which you want PDM to write the merge output file.
- Specify other options related to the merge process

Note: When PDM creates the merge output file automatically during the Build WIP process, the merge output file name is the same as the WIP member name. You cannot change the name of the Merge Output file.

When you press ENTER after providing information on this panel:

- The AllFusion Endevor Change Manager Parallel Development Option menu appears with a message indicating the merge process has succeeded or failed. The merge process fails if there are conflicts in the WIP file.
- The message log appears if the merge process fails for some other reason.
- The WIP Selection Matrix appears if Root or either of the Derivation files were not explicitly specified.

2.6.1 Merge Output Specification Panel Fields

Field	Description
Merge Library fields	Use to specify the merge library, using standard ISPF conventions. 1
Data Set Name	Use as an alternative to the MERGE LIBRARY fields.
If Partitioned, Replace Like-named Members	Indicates whether PDM is to replace like-named members in the merge output library. The default is Y.
Delete WIP Member after Merge	Tells PDM whether to delete the WIP member after creating the merge output file. The default is N.
Panvalet/ Librarian Language	Optional. This field associates a AllFusion CA-Panvalet or AllFusion CA-Librarian type identifier with the named output member. PDM ignores the field if the merge data set is not an AllFusion CA-Panvalet or an AllFusion CA-Librarian data set. Selections include: ▪ **AllFusion CA-Panvalet —** ALC, ANSCOBOL, AUTOCODE, BAL, COBOL, COBOL-72, DATA, FORTRAN, JCL, OBJECT, PL/1, RPG, USER180, USER780, OTHER. ▪ **AllFusion CA-Librarian —** ASM, COB, DAT, FOR, FRG, FRS, GIF, GOF, JCL, PLF, PLI, RPG, TXT, VSB. If left blank, the default value is DATA for AllFusion CA-Panvalet, and DAT for AllFusion CA-Librarian.

Note: 1 : The record length of the Merge data set must be large enough to support the largest WIP record. Generally this means that the Merge data set must be no shorter than eight bytes less than the WIP record length. For example, if the WIP data set record length is 88, then the Merge data set record length must be at least 80.

2.7 Using the WIP Member Selection Matrix

The WIP Member Selection Matrix contains a header identifying the WIP, Root, Derivation 1 and Derivation 2 libraries or Endevor locations. The matrix also displays the member or element names within the Root, Derivation 1, and Derivation 2 libraries or Endevor locations next to their corresponding WIP member names.

The only difference caused by designating Root, Derivation 1, and/or Derivation 2 from Endevor is in the format of the matrix header, as follows:

- An Endevor source name is defined using the environment, stage, system, subsystem, and type, separated by slashes.

```
Work-in-Process:  ENDEVOR.PDM.WIPLIB
Root:             SUPPORT /B /FINANCE /ACCTREC /COPYBOOK
Derivation 1:     SUPPORT /A /FINANCE /ACCTREC /COPYBOOK
Derivation 2:     DEMO    /P /FINANCE /ACCTREC /COPYBOOK
```

- A data set source name is defined using standard data set naming conventions.

```
Work-in-Process:  ENDEVOR.PDM.WIPLIB
Root:             ENDEVOR.PDM.ROOT
Derivation 1:     ENDEVOR.PDM.DV1LIB
Derivation 2:     ENDEVOR.PDM.DV2LIB
```

2.7.1 Sample Matrix

The matrix can be a useful tool because it allows you to view and manipulate matrix rows prior to invoking the Build WIP process.

```
Build WIP --------------- WIP Member Selection Matrix -------------------------
Command ===>                                                  Scroll ===>
  CREATE - Create WIP or SCL  Locate - Position matrix   PrtMat - Print matrix
  SORT - Sort matrix          QUIT - Quit matrix
Work-in-Process:  ENDEVOR.PDM.WIPLIB
Root:             ENDEVOR.PDM.ROOTLIB
Derivation 1:     ENDEVOR.PDM.DV1LIB
Derivation 2:     ENDEVOR.PDM.DV2LIB
Row Commands:  I Include   X Exclude   R Rename
   ---- WIP -----  Rename      --- Root ---- --- Deriv. 1 --- --- Deriv. 2 ---
  0001 WC1PAL00                 BC1PAL00         AC1PAL00         BC1PAL00
  0002 WC1PAL10                 BC1PAL10
  0003 WC1PBDWK                 BC1PBDWK                          BC1PBDWK
  0004 WC1PBM30                 BC1PBM30
  0005 WC1PBM35                 BC1PBM35
  0006 WC1PBM40                 BC1PBM40
  0007 WC1PBR10                 BC1PBR10
  0008 WC1PCAF                  BC1PCAF
  0009 WC1PCIOB                 BC1PCIOB         AC1PCIOB         BC1PCIOB
  0010 WC1PCONG                 BC1PCONG
  0011 WC1PCONP                 BC1PCONP
  0012 WC1PC1PR                 BC1PC1PR         AC1PC1PR         BC1PC1PR
```

Each row in the matrix consists of the name of a WIP file and of the Root and Derivation files from which it is built. Rows that contain the names of a WIP and Root are assigned a sequence number. Unmatched Derivation 1 or Derivation 2

members appear on the matrix after the last sequenced row, and are not assigned sequence numbers.

Use the PF8 and PF7 keys to scroll through the matrix. To find a particular WIP member within the matrix, you can use the locate command.

When you issue the CREATE command PDM processes all matrix rows containing:

- WIP, Root and Derivation 1 file specifications.
- WIP, Root, Derivation 1 and Derivation 2 file specifications.

Note: PDM does not process rows with only WIP, Root, and Derivation 2 file specifications.

For example, PDM processes rows 0001, 0009, and 0012. The remaining rows are not processed because they do not include a Derivation 1 file.

```
---- WIP -----  Rename      --- Root ---- --- Deriv. 1 --- --- Deriv. 2 ---
0001 WC1PAL00                BC1PAL00         AC1PAL00         BC1PAL00
0002 WC1PAL10                BC1PAL10
0003 WC1PBDWK                BC1PBDWK                         BC1PBDWK
0004 WC1PBM30                BC1PBM30
0005 WC1PBM35                BC1PBM35
0006 WC1PBM40                BC1PBM40
0007 WC1PBR10                BC1PBR10
0008 WC1PCAF                 BC1PCAF
0009 WC1PCIOB                BC1PCIOB         AC1PCIOB         BC1PCIOB
0010 WC1PCONG                BC1PCONG
0011 WC1PCONP                BC1PCONP
0012 WC1PC1PR                BC1PC1PR         AC1PC1PR         BC1PC1PR
```

The WIP Member Selection Matrix panel provides row commands that allow you to restrict the rows processed by PDM, by:

- Excluding rows from the Build WIP process.
- Including previously excluded rows in the Build WIP process.
- Renaming members that currently exist in the WIP data set.
- Moving any Derivation 1 or Derivation 2 members to match them with specific Root members.

The WIP Member Selection Matrix panel also provides commands for manipulating the entire matrix. Specifically, you can:

- Print the matrix, using the PRTMAT command.
- Position the matrix on a line using the LOCATE command.
- Sort the matrix, using the SORT command
- Create WIP files or build SCL using the CREATE command.

These row and matrix commands are discussed in the following sections.

2.7.2 Excluding a WIP Member

Use the EXCLUDE command to exclude a WIP matrix row from processing. You may decide that rather than rename a particular member, you will just not process it. Or, you may want to build WIP Files for only those matrix rows with members in the Root and at least one Derivation. The exclude command allows you to quickly eliminate unnecessary processing.

To exclude a WIP member(s) from processing, type **X** in the SELECTION field next to the WIP member(s) to be excluded.

```
Build WIP --------------- WIP Member Selection Matrix -----------------------
Option ===>                                                    Scroll ===>
  CREATE - Create matrix   Locate - Position matrix   PrtMat - Print matrix
  SORT - Sort matrix       QUIT - Quit matrix
Work-in-Process:  ENDEVOR.PDM.WIPLIB
Root:             ENDEVOR.PDM.ROOTLIB
Derivation 1:     ENDEVOR.PDM.DV1LIB
Derivation 2:     ENDEVOR.PDM.DV2LIB
Row Commands:  I Include   X Exclude   R Rename
   ---- WIP -----  Rename      --- Root ---- --- Deriv. 1 --- --- Deriv. 2 -
   0001 WC1PAL00                BC1PAL00         AC1PAL00         BC1PAL00
x  0002 WC1PAL10                BC1PAL10
   0003 WC1PBDWK                BC1PBDWK                          BC1PBDWK
x  0004 WC1PBM30                BC1PBM30
x  0005 WC1PBM35                BC1PBM35
x  0006 WC1PBM40                BC1PBM40
x  0007 WC1PBR10                BC1PBR10
x  0008 WC1PCAF                 BC1PCAF
   0009 WC1PCIOB                BC1PCIOB         AC1PCIOB         BC1PCIOB
x  0010 WC1PCONG                BC1PCONG
x  0011 WC1PCONP                BC1PCONP
   0012 WC1PC1PR                BC1PC1PR         AC1PC1PR         BC1PC1PR
```

Press ENTER. The screen returns with the message ***EXCLUDED** in the RENAME field, next to the appropriate WIP member(s). Note that there is no longer a sequence number for the member.

```
Build WIP --------------- WIP Member Selection Matrix -----------------------
Option ===>                                                    Scroll ===>
  CREATE - Create matrix   Locate - Position matrix   PrtMat - Print matrix
  SORT - Sort matrix       QUIT - Quit matrix
Work-in-Process:  ENDEVOR.PDM.WIPLIB
Root:             ENDEVOR.PDM.ROOTLIB
Derivation 1:     ENDEVOR.PDM.DV1LIB
Derivation 2:     ENDEVOR.PDM.DV2LIB
Row Commands:  I Include   X Exclude   R Rename
   ---- WIP -----  Rename      --- Root ---- --- Deriv. 1 --- --- Deriv. 2 -
  0001 WC1PAL00                BC1PAL00         AC1PAL00         BC1PAL00
       WC1PAL10   *EXCLUDED    BC1PAL10
  0003 WC1PBDWK                BC1PBDWK                          BC1PBDWK
       WC1PBM30   *EXCLUDED    BC1PBM30
       WC1PBM35   *EXCLUDED    BC1PBM35
       WC1PBM40   *EXCLUDED    BC1PBM40
       WC1PBR10   *EXCLUDED    BC1PBR10
       WC1PCAF    *EXCLUDED    BC1PCAF
  0009 WC1PCIOB                BC1PCIOB         AC1PCIOB         BC1PCIOB
       WC1PCONG   *EXCLUDED    BC1PCONG
       WC1PCONP   *EXCLUDED    BC1PCONP
  0012 WC1PC1PR                BC1PC1PR         AC1PC1PR         BC1PC1PR
```

2.7.3 Including Matrix Rows

The INCLUDE command can be used to include a matrix row that you have previously excluded, or to override the "no replace" parameter.

To include a matrix row, type **I** next to the appropriate member(s), then press ENTER.

```
Build WIP --------------- WIP Member Selection Matrix -----------------------
Option ===>                                                    Scroll ===>
  CREATE - Create matrix   Locate - Position matrix   PrtMat - Print matrix
  SORT - Sort matrix       QUIT - Quit matrix
Row Commands:  I Include   X Exclude   R Rename
  ---- WIP -----  Rename      --- Root ---- --- Deriv. 1 --- --- Deriv. 2 -
  0001 WC1PAL00                BC1PAL00         AC1PAL00         BC1PAL00
  0002 WC1PAL10                BC1PAL10
I 0003 WC1PBDWK                BC1PBDWK                          BC1PBDWK
  0004 WC1PBM30                BC1PBM30
  0005 WC1PBM35                BC1PBM35
I 0006 WC1PBM40                BC1PBM40
  0007 WC1PBR10                BC1PBR10
  0008 WC1PCAF                 BC1PCAF
  0009 WC1PCIOB                BC1PCIOB         AC1PCIOB         BC1PCIOB
I 0010 WC1PCONG                BC1PCONG
  0011 WC1PCONP                BC1PCONP
  0012 WC1PC1PR                BC1PC1PR         AC1PC1PR         BC1PC1PR
```

The screen returns with the message ***INCLUDED** in the RENAME field, next to the appropriate WIP member(s), to indicate that the member has been included. A sequence number also appears for the member.

```
Build WIP --------------- WIP Member Selection Matrix -----------------------
Option ===>                                                    Scroll ===>
  CREATE - Create matrix   Locate - Position matrix   PrtMat - Print matrix
  SORT - Sort matrix       QUIT - Quit matrix
Work-in-Process:  ENDEVOR.PDM.WIPLIB
Root:             ENDEVOR.PDM.ROOTLIB
Derivation 1:     ENDEVOR.PDM.DV1LIB
Derivation 2:     ENDEVOR.PDM.DV2LIB
Row Commands:  I Include   X Exclude   R Rename
  ---- WIP -----  Rename      --- Root ---- --- Deriv. 1 --- --- Deriv. 2 -
  0001 WC1PAL00                BC1PAL00         AC1PAL00         BC1PAL00
  0002 WC1PAL10                BC1PAL10
  0003 WC1PBDWK   *INCLUDED    BC1PBDWK                          BC1PBDWK
  0004 WC1PBM30                BC1PBM30
  0005 WC1PBM35                BC1PBM35
  0006 WC1PBM40   *INCLUDED    BC1PBM40
  0007 WC1PBR10                BC1PBR10
  0008 WC1PCAF                 BC1PCAF
  0009 WC1PCIOB                BC1PCIOB         AC1PCIOB         BC1PCIOB
  0010 WC1PCONG   *INCLUDED    BC1PCONG
  0011 WC1PCONP                BC1PCONP
  0012 WC1PC1PR                BC1PC1PR         AC1PC1PR         BC1PC1PR
```

2.7.4 Renaming a WIP Member

Use the RENAME command to assign a new name to a member with the same name as another member in the WIP data set.

To rename a member, type an **R** in the SELECTION field next to the member(s) to be renamed, type the new name in the RENAME field, then press ENTER.

```
Build WIP --------------- WIP Member Selection Matrix -----------------------
Option ===>                                                    Scroll ===>
  CREATE - Create matrix   Locate - Position matrix   PrtMat - Print matrix
  SORT - Sort matrix       QUIT - Quit matrix
Work-in-Process:  ENDEVOR.PDM.WIPLIB
Root:             ENDEVOR.PDM.ROOTLIB
Derivation 1:     ENDEVOR.PDM.DV1LIB
Derivation 2:     ENDEVOR.PDM.DV2LIB
Row Commands:  I Include   X Exclude   R Rename
  ---- WIP -----  Rename      --- Root ---- --- Deriv. 1 --- --- Deriv. 2 -
  0001 WC1PAL00                BC1PAL00         AC1PAL00         BC1PAL00
r 0002 WC1PAL10   WIPPA1110    BC1PAL10
  0003 WC1PBDWK                BC1PBDWK                          BC1PBDWK
  0004 WC1PBM30                BC1PBM30
  0005 WC1PBM35                BC1PBM35
  0006 WC1PBM40                BC1PBM40
  0007 WC1PBR10                BC1PBR10
  0008 WC1PCAF                 BC1PCAF
  0009 WC1PCIOB                BC1PCIOB         AC1PCIOB         BC1PCIOB
  0010 WC1PCONG                BC1PCONG
  0011 WC1PCONP                BC1PCONP
  0012 WC1PC1PR                BC1PC1PR         AC1PC1PR         BC1PC1PR
```

The screen returns with the message ***RENAMED** in the RENAME field, next to the new member name.

```
Build WIP --------------- WIP Member Selection Matrix -----------------------
Option ===>                                                    Scroll ===>
  CREATE - Create matrix   Locate - Position matrix   PrtMat - Print matrix
  SORT - Sort matrix       QUIT - Quit matrix
Work-in-Process:  ENDEVOR.PDM.WIPLIB
Root:             ENDEVOR.PDM.ROOTLIB
Derivation 1:     ENDEVOR.PDM.DV1LIB
Derivation 2:     ENDEVOR.PDM.DV2LIB
Row Commands:  I Include   X Exclude   R Rename
   ---- WIP -----  Rename      --- Root ---- --- Deriv. 1 --- --- Deriv. 2 -
  0001 WC1PAL00                BC1PAL00         AC1PAL00         BC1PAL00
  0002 WIPPAL10   *RENAMED     BC1PAL10
  0003 WC1PBDWK                BC1PBDWK                          BC1PBDWK
  0004 WC1PBM30                BC1PBM30
  0005 WC1PBM35                BC1PBM35
  0006 WC1PBM40                BC1PBM40
  0007 WC1PBR10                BC1PBR10
  0008 WC1PCAF                 BC1PCAF
  0009 WC1PCIOB                BC1PCIOB         AC1PCIOB         BC1PCIOB
  0010 WC1PCONG                BC1PCONG
  0011 WC1PCONP                BC1PCONP
  0012 WC1PC1PR                BC1PC1PR         AC1PC1PR         BC1PC1PR
```

2.7.5 Moving a Derivation 1 or Derivation 2 Member

Unmatched Derivation 1 and Derivation 2 member names are listed at the end of the WIP Member Selection Matrix. You may realize that some of these members actually do match listed Root members. Or, you may want to replace a Derivation 1 or Derivation 2 member in a matrix row with a different member. PDM allows you to move Derivation 1 and Derivation 2 members into the different matrix rows.

To move the member, tab to the member name you want to move. Type the number of the matrix row to which you want to move this member before the member name, then press ENTER. If the member being moved is replacing an existing member, the old member is moved to the first available open slot in the matrix, after the last matrix row with a sequence number.

In the example below, to replace the Derivation 1 member in the matrix row number 0352 (A1SV1000) with the Derivation 1 member AMDV1000, type the matrix row number (0352) next to that Derivation 1 member (AMDV1000).

```
Build WIP --------------- WIP Member Selection Matrix -----------------------
Option ===>                                                  Scroll ===>
  CREATE - Create matrix   Locate - Position matrix   PrtMat - Print matrix
  SORT - Sort matrix       QUIT - Quit matrix
Work-in-Process:  ENDEVOR.PDM.WIPLIB
Root:             ENDEVOR.PDM.ROOTLIB
Derivation 1:     ENDEVOR.PDM.DV1LIB
Derivation 2:     ENDEVOR.PDM.DV2LIB
Row Commands:  I Include   X Exclude   R Rename
   ---- WIP -----  Rename      --- Root ---- --- Deriv. 1 --- --- Deriv. 2 -
  0351 W1SSXI10                C1SSXI10         A1SSXI10
  0352 W1SV1000                C1SV1000         A1SV1000
  0353 W1UEXITS                C1UEXITS
  0354 WPPROCSS                DPPROCSS
  0355 WPC1UEXT                EPC1UEXT
  0356 WPPROCSS                GPPROCSS
  0357 WSWASM01                JSWASM01         ASWASM01
  0358 WSWASM02                JSWASM02
  0359 WSWASM03                JSWASM03         ASWASM03
  0360 WROGX                   PROGX
  0361 X                       X                              X
                                          0352 AMDV1000       @TODO
```

When you press ENTER, the Derivation 1 member AMDV1000 is moved to matrix row number 0352, replacing an existing member (A1SV1000), and member A1SV1000 is moved to the first available open slot in the matrix.

```
Build WIP --------------- WIP Member Selection Matrix -----------------------
Option ===>                                                  Scroll ===>
  CREATE - Create matrix   Locate - Position matrix   PrtMat - Print matrix
  SORT - Sort matrix       QUIT - Quit matrix
Work-in-Process:  ENDEVOR.PDM.WIPLIB
Root:             ENDEVOR.PDM.ROOTLIB
Derivation 1:     ENDEVOR.PDM.DV1LIB
Derivation 2:     ENDEVOR.PDM.DV2LIB
Row Commands:  I Include   X Exclude   R Rename
   ---- WIP -----  Rename      --- Root ---- --- Deriv. 1 --- --- Deriv. 2 -
  0351 W1SSXI10                C1SSXI10         A1SSXI10
  0352 W1SV1000                C1SV1000         AMDV1000
  0353 W1UEXITS                C1UEXITS
  0354 WPPROCSS                DPPROCSS
  0355 WPC1UEXT                EPC1UEXT
  0356 WPPROCSS                GPPROCSS
  0357 WSWASM01                JSWASM01         ASWASM01
  0358 WSWASM02                JSWASM02
  0359 WSWASM03                JSWASM03         ASWASM03
  0360 WROGX                   PROGX
  0361 X                       X                              X
                                                               @TODO
                                          A1SV1000
```

2.7.6 Like-Named WIP Members

Occasionally, you may specify a WIP data set that contains a member with the same name as another member. If you answered **Y** for the IF PARTITIONED, REPLACE LIKE-NAMED MEMBERS option on the WIP Specification Panel, PDM replaces the existing member with the new member. If you indicated **N** for the option, the like-named member is not processed.

```
Build WIP --------------- WIP Member Selection Matrix -----------------------
Option ===>                                                    Scroll ===>
  CREATE - Create matrix   Locate - Position matrix   PrtMat - Print matrix
  SORT - Sort matrix       QUIT - Quit matrix
Work-in-Process:  ENDEVOR.PDM.WIPLIB
Root:             ENDEVOR.PDM.ROOTLIB
Derivation 1:     ENDEVOR.PDM.DV1LIB
Derivation 2:     ENDEVOR.PDM.DV2LIB
Row Commands:  I Include   X Exclude   R Rename
   ---- WIP -----  Rename      --- Root ---- --- Deriv. 1 --- --- Deriv. 2 -
  0001 WC1PAL00                BC1PAL00         AC1PAL00         BC1PAL00
       WC1PAL10   *NO-REPL     BC1PAL10
  0003 WC1PBDWK                BC1PBDWK                          BC1PBDWK
  0004 WC1PBM30                BC1PBM30
  0005 WC1PBM35                BC1PBM35
  0006 WC1PBM40                BC1PBM40
  0007 WC1PBR10                BC1PBR10
  0008 WC1PCAF                 BC1PCAF
  0009 WC1PCIOB                BC1PCIOB         AC1PCIOB         BC1PCIOB
  0010 WC1PCONG                BC1PCONG
  0011 WC1PCONP                BC1PCONP
  0012 WC1PC1PR                BC1PC1PR         AC1PC1PR         BC1PC1PR
```

The message ***NO-REPL** in the RENAME column indicates that the data set ENDEVOR.PDM.WIPLIB already contains the member WC1PAL10, and that the IF PARTITIONED, REPLACE LIKE-NAMED MEMBERS option was not chosen (that is, **N** was entered in this field).

In this situation, you have three options:

- You can leave the member name as is, which means that the member is not processed.
- If you want to replace this particular member in the WIP data set and you are in foreground mode, you can simply *include* the member. The INCLUDE command is discussed earlier in this section.
- If you want to keep both like-named members and/or you are processing in batch mode, you must *include and rename* the member. The RENAME command is also discussed earlier in this section.

```
 Build WIP --------------- WIP Member Selection Matrix -----------------------
Option ===>                                                      Scroll ===>
  CREATE - Create matrix   Locate - Position matrix   PrtMat - Print matrix
  SORT - Sort matrix       QUIT - Quit matrix
Work-in-Process:  ENDEVOR.PDM.WIPLIB
Root:             ENDEVOR.PDM.ROOTLIB
Derivation 1:     ENDEVOR.PDM.DV1LIB
Derivation 2:     ENDEVOR.PDM.DV2LIB
Row Commands:  I Include   X Exclude   R Rename
   ---- WIP -----  Rename      --- Root ---- --- Deriv. 1 --- --- Deriv. 2 -
  0001 WC1PAL00                BC1PAL00         AC1PAL00         BC1PAL00
  0002 WIPPAL10   *RENAMED     BC1PAL10
  0003 WC1PBDWK                BC1PBDWK                          BC1PBDWK
  0004 WC1PBM30                BC1PBM30
  0005 WC1PBM35                BC1PBM35
  0006 WC1PBM40                BC1PBM40
  0007 WC1PBR10                BC1PBR10
  0008 WC1PCAF                 BC1PCAF
  0009 WC1PCIOB                BC1PCIOB         AC1PCIOB         BC1PCIOB
  0010 WC1PCONG                BC1PCONG
  0011 WC1PCONP                BC1PCONP
  0012 WC1PC1PR                BC1PC1PR         AC1PC1PR         BC1PC1PR
```

2.7.7 Printing the Matrix

The PRTMAT command syntax is:

```
PRTMAT sysout-class destination
```

PRTMAT
Prints a formatted version of the matrix.

sysout-class
An optional parameter that identifies the output class. If the output class is not specified, PDM uses the values specified on the PDM User Defaults panel.

destination
An optional parameter that identifies the output destination. If the destination is not specified, PDM uses the values specified on the PDM User Defaults panel.

Note: If no defaults are provided, PDM routes the output to SYSOUT class A.

Example

Command	Description
PRTMAT C	Prints the matrix to class C.
PRTMAT C PRNT15	Prints the matrix to class C printer number 15.

2.7.8 Positioning the Matrix

The LOCATE command syntax is:

```
LOCATE  row-value  column-name
```

Locate
Positions the matrix to a specified row value. This command can be abbreviated to *loc* or *l*

row-value
Identifies the value being used in the search.

column-name
This optional parameter identifies the column being searched. The default LOCATE column is the WIP member name. If a SORT command is issued, the column name parameter provided becomes the default LOCATE column. Valid values include:

- WIPSEQ — WIP sequence number
- WIP — WIP member name
- ROOT — Root member name
- DER1 — Derivation 1 member name
- DER2 — Derivation 2 member name

Notes:

If the column name specified in the LOCATE command is not the current sort column, the LOCATE command may appear to position the matrix to the incorrect row. This can occur if:

- The column name specified is not sorted.
- The column contains renamed members.
- The column contains unmatched members.

To correctly position the matrix in these instances, use the SORT command to sort the matrix.

Example

Command	Description
LOC PROGX	Positions the matrix at the first row containing PROGX.
L PROGX DER1	Positions the matrix at the first row containing PROGX in the DER1 column.

2.7.9 Sorting the Matrix

The SORT command syntax is:

```
SORT column-name
```

SORT
Orders the matrix into ascending sequence based on the value of column-name.

column-name
Identifies which column to sort into ascending order. The default column is the WIP member name. Valid values include:

- WIPSEQ — WIP sequence number.
- WIP — WIP member name.
- ROOT — Root member name
- DER1 — Derivation 1 member name
- DER2 — Derivation 2 member name.

Example

Command	Description
SORT ROOT	Sorts the matrix based on the Root column. The Root column now becomes the default LOCATE column.

2.7.10 Creating WIP Files or SCL

Once you have edited the matrix to meet all your requirements, you can create WIP files or SCL for each member or element. To do this, type **CREATE** in the COMMAND field, then press ENTER.

Note: When you issue the CREATE command, PDM only processes included rows that have at least a Root and Derivation 1 member.

```
Build WIP --------------- WIP Member Selection Matrix -----------------------
Command ===> create                                              Scroll ===>
  CREATE - Create matrix   Locate - Position matrix   PrtMat - Print matrix
  SORT - Sort matrix       QUIT - Quit matrix
Work-in-Process:  ENDEVOR.PDM.WIPLIB
Root:             ENDEVOR.PDM.ROOTLIB
Derivation 1:     ENDEVOR.PDM.DV1LIB
Derivation 2:     ENDEVOR.PDM.DV2LIB
Row Commands:  I Include   X Exclude   R Rename
     ---- WIP -----  Rename      --- Root ---- --- Deriv. 1 --- --- Deriv. 2 -
    0001 WC1PAL00                BC1PAL00         AC1PAL00         BC1PAL00
    0002 WIPPAL10                BC1PAL10
    0003 WC1PBDWK                BC1PBDWK                          BC1PBDWK
    0004 WC1PBM30                BC1PBM30
    0005 WC1PBM35                BC1PBM35
    0006 WC1PBM40                BC1PBM40
    0007 WC1PBR10                BC1PBR10
    0008 WC1PCAF                 BC1PCAF
    0009 WC1PCIOB                BC1PCIOB         AC1PCIOB         BC1PCIOB
    0010 WC1PCONG                BC1PCONG
    0011 WC1PCONP                BC1PCONP
    0012 WC1PC1PR                BC1PC1PR         AC1PC1PR         BC1PC1PR
```

If you are in foreground mode, PDM begins the Build WIP process for all selected rows. If you are in batch mode, PDM creates batch requests for all selected rows and writes the requests to the batch request data set. The batch files can be processed with the Submit option on the Primary Option Menu.

To cancel the Build WIP process, type **QUIT** in the OPTION field and press ENTER to return to the AllFusion Endevor Change Manager Parallel Development Option menu

When the Build WIP process is complete, the AllFusion Endevor Change Manager Parallel Development Option menu returns with one of these messages:

- **WIP Build Complete** — If you are processing in foreground mode
- **Requests Written** — If you are processing in batch mode

2.8 Using the Summary of Levels Panel

When using an Endevor location as the Build WIP input location, you can tell PDM to use a level other than the current version and level during the build operation.

If you know the version and level you want to use, you can enter them directly on the Root, Derivation 1, or Derivation 2 Specification Panel. Otherwise, you can review a Summary of Levels panel to find the version/level you want to use. To access a Summary of Levels panel:

- From a specification panel, type **S** in the COMMAND field, an explicit name or name mask in the ELEMENT field, then press ENTER.
  - If you provide an explicit element name the Summary of Levels panel displays.
  - If you provide a mask, an Element Selection List displays. Select an element from this list, then press ENTER to display the Summary of Levels panel.
- From an Element Selection List, type **S** in the COMMAND field, **S** to the desired element, then press ENTER.

```
BUILD WIP  ------------------- SUMMARY OF LEVELS ----------- ROW 1 TO 4 OF 4
 COMMAND ===>                                                SCROLL ===> PAGE
   Environment: SMPLTEST           System: ADMIN            Subsystem: PROCESS
   Element:     C1SDTY00           Type:   ISPPE            Stage:     T
 ------------------------  SOURCE LEVEL INFORMATION  --------------------
     VV.LL   USER       DATE      TIME    STMTS     INS     DEL   CCID
     01.00   ZSXSXV12   04APR02   20:15      27       0       0   RELEASE 15
     01.01   ZSXSXV11   26MAY02   07:58      27       2       2   HELP
     01.02   ZSXSXV1A   23AUG02   14:48      27       1       1   TUTORIALS
     01.03   NA1JM44    30OCT02   08:36      27       1       1   36-0949
 ****************************** BOTTOM OF DATA ****************************
```

Note: To find a particular version/level within the Summary of Levels display, you can use the ISPF locate command. The command must be specified in the COMMAND field, and pertains to version and level only. You can either enter the full command (locate 01.01) or abbreviate it (loc 01.01 or l 01.01).

2.8.1 Using the Summary of Levels Panel with the WIP Member Selection Matrix

You can also access the Summary of Levels panel from the WIP Member Selection Matrix. Type an **S** next to the element name for which you want to use a particular version and level. Press ENTER; the Summary of Levels screen is returned.

Select the version and level you want to use from the Summary of Levels screen, as described in the preceding section. Press ENTER; the WIP Member Selection Matrix reappears.

In the example below, a specific version/level has been selected for the element COPYRECB. The "**<**" appended to the end of the element name indicates a version and level other than the current version/level is used for this element when building the WIP File.

```
Build WIP ----------- WIP Member Selection Matrix     LINK 000001 COL 001 080
Option ===>                                                  Scroll ===>
  CREATE - Create matrix   Locate - Position matrix   PrtMat - Print matrix
  SORT - Sort matrix       QUIT - Quit matrix
Work-in-Process:  ENDEVOR.PDM.WIPLIB
Root:             SUPPORT /B /FINANCE /ACCTREC /COPYBOOK
Derivation 1:     SUPPORT /A /FINANCE /ACCTREC /COPYBOOK
Derivation 2:     DEMO    /P /FINANCE /ACCTREC /COPYBOOK
Row Commands:  I Include   X Exclude   R Rename
   ---- WIP ----  Rename      --- Root ---- --- Deriv. 1 --- --- Deriv. 2 --
  0001 COPYRECA                COPYRECA
  0002 COPYRECB                COPYRECB  <
  0003 COPYRECC                COPYRECC
  0004 COPY1                   COPY1            COPY1            COPY1
  0005 COPY2                   COPY2            COPY2            COPY2
  0006 C1CLINK                 C1CLINK          C1CLINK          C1CLINK
  0007 C1DEMOCB                C1DEMOCB         C1DEMOCB         C1DEMOCB
  0008 HEADER1                 HEADER1                           HEADER1
  0009 PAGING                  PAGING                            PAGING
                                                DSNRECD
                                                ELEMENT
```

2.8.1.1 Clearing an Assigned Version/Level

You can clear the version and level designated for an element by blanking out the entries on either the specification panel on which they appear or the Summary of Levels screen.

To clear the version and level on the specification panel, tab to the VERSION and LEVEL fields and space through the entries. The current version and level will now be used for the element. If you want to assign a different version and level, type the new entries on this panel or use the Summary of Levels panel.

To clear the version and level on the Summary of Levels panel, tab to the version/level displayed next to the element name (in the top portion of the screen) and space over the entry. Press ENTER.

```
BUILD WIP ------------------ SUMMARY OF LEVELS -------------- ROW 1 OF 64
 COMMAND ===>                                                 SCROLL ===> PAGE
   Environment: SMPLTEST          System: ADMIN              Subsystem: PROCESS
   Element:    COPYRECB   01.00   Type:   COPYBOOK         Stage:     T
 ----------------------  SOURCE LEVEL INFORMATION  -------------------------
     VV.LL   USER       DATE      TIME    STMTS     INS     DEL   CCID
     01.00   ZSXSXV11   08APR02   10:40     691       0       0   RELEASE 15
     01.01   ZSXSXV11   08APR02   12:46     774      97      14   RELEASE 15
     01.02   ZSXPGM1    11APR02   01:57     776       3       1   RELEASE 15
     01.03   ZSXSXV11   20APR02   17:45     787      17       6   S.VELLOFF
     01.04   ZSXDBH11   05MAY02   16:15     793       9       3
     01.05   ZSXPGM1    13MAY02   15:52     796      18      15   CONDORO3
     01.06   ZSXPGM1    13MAY02   15:57     795       2       3   CONDORO3
     01.07   ZSXLDG1    24MAY02   18:43     799       4       0   FORCE
     01.08   ZSXPGM1F   10JUN02   10:34     804      18      13   LIST
     01.09   ZSXPGM14   17JUN02   18:33     825      61      40
     01.10   ZSXBAP1X   23JUN02   17:09     828       7       4   SCL
     01.11   ZSXPGM1X   24JUN02   14:10     840      12       0   TRANSFER
     01.12   ZSXBAP1J   24JUN02   16:57     844       4       0   SCL
     01.13   ZSXPGM1I   25JUN02   15:49     844       4       4   TRANSFER
     01.14   ZSXSXV1    28JUN02   12:34     846       4       2   QA0214
     01.15   ZSXSXV1A   28JUN02   18:31     848       2       0   QA0214
```

The Summary of Levels screen reappears, with no version/level for the element. The current version and level will be used automatically for the element, unless you select a different version/level from the panel. You must press PF3 to return to your previous screen.

Once you clear the version and level for the element, the "<" no longer appears at the end of the element name on the WIP Member Selection Matrix.

2.9 The Next Step

Building the WIP File (or WIP File requests) and analyzing the results is the first step in the PDM process. You can review the WIP Summary Report during the Build WIP process without actually building the WIP file. Refer to the sections Specifying a WIP Library and Build WIP Options earlier in this chapter.

The PDM reports, listed below, can be generated during Build WIP processing and are discussed in detail in Chapter 6, “PDM Reports”:

- PDM Syntax Report
- PDM Build WIP Detail Report
- PDM Build WIP Summary Report

The next step is to consolidate and resolve conflicts and duplicate code by editing the WIP File. Use the PDM edit WIP function to make the appropriate changes to the WIP file.

Chapter 3. Editing WIP Files

3.1 The Format of the WIP File

PDM builds the WIP File by adding, and annotating, all the changes made in the Derivation 1 and Derivation 2 files to the Root file. It then writes this new file to the WIP library. The WIP file is made up of:

- Header information
- WIP file annotations
- The WIP file itself

This section describes these components.

3.1.1 The WIP File Header

```
000001 *************** PARALLEL DEVELOPMENT Option WIP SUMMARY ******** ****
000002 **                                                                  *
000003 ** ROOT:         ENDEVOR.PDM.ROOT                                  *
000004 ** DERIVATION 1: ENDEVOR.PDM.DV1                                   *
000005 ** DERIVATION 2: ENDEVOR.PDM.DV2                                   *
000006 **                                                                  *
000007 ** ROOT:PROG3200          DER1:PROG3200          DER2:PROG3200      *
000008 **                                                                  *
000009 ** NUMBER OF WIP RECORDS......    3784                             *
000010 ** NUMBER OF DELETED RECORDS...     35,DER1     0,DER2    32,COM      3 *
000011 ** NUMBER OF INSERTED RECORDS..     63,DER1     0,DER2    45,COM     18 *
000012 ** NUMBER OF CONFLICTS.....   0                                     *
000013 ** COMPLEXITY FACTOR..... 1                                        *
000014 ** BUILD RETURN CODE.....  0                                       *
000015 **                                                                  *
000016 ******* ----+----1----+----2----+----3----+----4----+----5----+----6----
```

3.1.1.1 WIP File Header Field Descriptions

Field	Description
Number of WIP Records	The number of lines in the WIP file.
Number of Deleted Records	The total number of lines flagged as deleted. The number of the deletions in DER1, in DER2, and common deletions are noted to the right of this figure.
Number of Inserted Records	The total number of lines flagged as inserted. The number of the insertions in DER1, in DER2, and common insertions are noted to the right of this figure.
Number of Conflicts	The number of areas where conflicting changes were found.
Complexity Factor	The PDM-calculated estimate of the severity of the conflicts in the WIP file.
Build Return Code	The highest return code from the Build WIP process.

3.1.2 The WIP File

A portion of a WIP file is shown below.

```
               ENVIRONMENT DIVISION.
               INPUT-OUTPUT SECTION.
               FILE-CONTROL.
                   SELECT REPORT-FILE ASSIGN U-T-SYSOUT.
                   SELECT INPUT-FILE ASSIGN U-T-INPUT.
               DATA DIVISION.
               FILE SECTION.
               FD REPORT-FILE
                   LABEL RECORDS ARE OMITTED
                   RECORDING MODE IS F
                   RECORD CONTAINS 133 CHARACTERS
                   DATA RECORD IS OUTPUT-RECORD.
                 01 OUTPUT-RECORD.
                   02 CARRIAGE-CONTROL PIC X.
                   02 OUTPUT-LINE      PIC X(132).
               FD INPUT-FILE
                   LABEL RECORDS ARE STANDARD
                   RECORD CONTAINS 80 CHARACTERS
                   DATA RECORD IS INPUT-FILE-RECORD.
                 01 INPUT-FILE-RECORD.
                   02 INPUT-FIELD1           PIC 9(8).
%?I-1              02 INPUT-FIELD2           PIC X(8).
%?I-2              02 INPUT-FIELD2           PIC X(6).
% D-1,2            02 INPUT-FIELD2           PIC X(10).
                   02 INPUT-FIELD3           PIC X(20).
                   02 INPUT-FIELD4           PIC 9(6).
                   02 INPUT-FIELD5           PIC 9(6).
%?I-1              02 FILLER                 PIC X(32).
%?I-2              02 FILLER                 PIC X(34).
```

As you review the WIP File, you notice various lines are annotated on the left side--in the first eight columns of the file. These annotations indicate the changes made to the Root program by each Derivation.

WIP File Annotations

Annotation	Description
%	Indicates that a change has occurred between the Root and one of the Derivations.
?	Indicates that a conflict exists at this point; this means that both Derivations made different changes to this line of the Root file.
I	Indicates that a line was added (inserted) by one of the Derivation files.
D	Indicates that a line was deleted by one of the Derivation files.
1 or 2	Indicates which Derivation the change came from, either 1 or 2. If the change came from both derivation files, the annotation would be 1,2.

Annotation	Description
*	Indicates a comment line. One or more asterisks may appear at the beginning of the line, but an asterisk in the first position of the line signifies that it is a comment line. PDM merge processing ignores comment lines.

Examples

This Annotation	Indicates That
% I-1	A line has been inserted by Derivation 1.
% D-2	A line has been deleted by Derivation 2.
% D-1,2	There has been a common deletion; both Derivation files deleted the same line.
%?I-1 %?I-2	A potential conflict exists at this point in the original program. Both programs inserted different lines at the same place.

In the example in 3.1.2, “The WIP File” on page 3-3, there are two instances of potential conflict, caused by two lines being inserted in the same place of the original program, and one instance of a deletion from both Derivation 1 and Derivation 2.

See 3.2, “Editing the WIP File” on page 3-5 for instructions on editing a WIP file.

3.2 Editing the WIP File

Once the WIP File has been built, you can edit it using the PDM Edit WIP function. The (edited) WIP File will be used as input to the Merge process, the result of which is an output file that can be added back into Endevor, stored in a source repository, or serve as input to a compiler.

To edit a WIP file, do the following:

1. Type **2** (EDIT WIP) in the OPTION field on the AllFusion Endevor Change Manager Parallel Development Option menu:

```
-------  AllFusion Endevor Change Manager Parallel Development Option  --------
 Option ===>  2

    0  DEFAULTS  - Specify user session parameters
    1  BUILD WIP - Build Work-in-Process (WIP) file
    2  EDIT WIP  - Edit Work-in-Process (WIP) file
    3  MERGE     - Merge Work-in-Process (WIP) file into OUTPUT source file
    4  SUBMIT    - Build job for batch submission
    T  TUTORIAL  - Tutorial on how to use the Parallel Development Option
    C  CHANGES   - New features for this release

Processing Mode:
   Foreground or Batch ===> F  (Foreground/Batch)

---------------------------- Batch Specification --------------------
Batch Request Data Set:               Batch Options:
   Project  ===>                         Append or Replace ===>     (A/R)
   Group    ===>
   Type     ===>
   Member   ===>
 Other partitioned or sequential data set:
   Data Set Name ===>

            (C) 1987,2002 Computer Associates International, Inc.
```

2. Press ENTER to display the Edit - Entry Panel. Specify the WIP library and the member you want to edit on this panel, using standard ISPF conventions.

```
  Edit WIP -------------------- Edit - Entry Panel -------------------------
  Command ===>
  ISPF Library:
     Project ===>  ENDEVOR
     Group   ===>  PDM
     Type    ===>  WIPLIB
     Member  ===>  PROGX           (Blank or pattern for member selection list)
  Other partitioned or sequential data set:
     Data Set Name  ===>
```

3. When you have specified the member you want to edit, press ENTER to display an ISPF/PDF edit panel.

```
EDIT ---- ENDEVOR.NDVR.WIPLIB(PROGX) - 01.00 ---------------------- COLUMNS 001 072
COMMAND ===>                                                 SCROLL ===> CSR
****** ***************************** TOP OF DATA ******************************
000001 *************** Parallel DEVELOPMENT Option WIP SUMMARY ******** ****
000002 **                                                                      *
000003 ** ROOT:          ENDEVOR.PDM.ROOT                                      *
000004 ** DERIVATION 1: ENDEVOR.PDM.DV1                                       *
000005 ** DERIVATION 2: ENDEVOR.PDM.DV2                                       *
000006 **                                                                      *
000007 ** ROOT:JEM2              DER1:JEM2              DER2:JEM2             *
000008 **                                                                      *
000009 ** NUMBER OF WIP RECORDS........      25                                *
000010 ** NUMBER OF DELETED RECORDS....       1  DER1:       0  DER2:       0 *
000011 ** NUMBER OF INSERTED RECORDS...       2  DER1:       1  DER2:       1 *
000012 ** NUMBER OF CONFLICTS..............   1                                 *
000013 ** COMPLEXITY FACTOR.................. 2                                 *
000014 ** BUILD RETURN CODE.................  0                                 *
000015 **                                                                      *
000016 ******* ----+----1----+----2----+----3----+----4----+----5----+----6----
000017                IDENTIFICATION DIVISION.
000018                PROGRAM-ID. PROGX.
000019 %?I-1          REMARKS. PDM TEST PROGRAM.
```

PDM invokes the standard ISPF/PDF editor. All ISPF/PDF edit commands are available. Resolve the conflicts in the WIP File, using these ISPF/PDF editing tools.

Note: Remember that the WIP File contains all insertions and deletions made by both Derivations, and flags any potential conflicts. If you decide not to edit the WIP File, PDM inserts all lines marked as insertions and deletes all lines marked as deletions during the Merge process. Because this could result in unresolved conflicts within the file, we recommend that you edit any WIP File member with a complexity factor higher than zero.

PDM provides commands, implemented as ISPF/PDF edit macros, to help you edit WIP files. These commands are described in the next section. See Appendix C, “Extending the WIP Edit Macro Facility” for more information on these commands.

3.3 WIP Edit Commands

To help edit the WIP file, PDM includes several commands that you can invoke from the ISPF/PDF Edit command line.

Note: To use these commands you must have accessed the WIP file using option 2 on the AllFusion Endevor Change Manager Parallel Development Option menu. These commands are not available if you edit the WIP file through the ISPF/PDF edit option.

3.3.1.1 WIP Edit Command Syntax

The syntax for invoking these commands is:

```
COMMAND-NAME  parameter1  parameter2  parameter3
```

3.3.1.2 WIP Edit Parameters

COMMAND-NAME
Valid values are:

WIPCOUNT
Counts the number of line records in the WIP member from the Root, Derivation 1, and Derivation 2 files.

WIPLDEL
Logically deletes lines from a specified WIP member.

WIPUNDEL
Restores any lines logically deleted by WIPLDEL, next it automatically invokes WIPCOUNT to recount the WIP statistics.

WIPSHOW
Displays only the lines that have not been logically deleted from the WIP member.

WIPCHANG
Displays only the lines that have changed during the WIP edit session.

WIPPARA
Identifies each inserted or deleted paragraph within the WIP member.

WIPCON
Displays only those lines identified as conflicts within the WIP member.

WIPMERGE
Creates a temporary Merge file from the WIP file.

WIPHELP
Displays help and tutorial information about the available WIP edit commands.

parameter*1* ... **parameter***n*
When a command can use parameters, the acceptable values of the parameters are:

R
To specify a Root file

1
To specify a Derivation 1 file

2
To specify a Derivation 2 file

3.3.1.3 WIP Edit Examples

The examples below provide an overview of the WIP edit command's general functionality. For more information individual commands:

- WIPCOUNT - see 3.3.2, “WIPCOUNT”
- WIPLDEL - see 3.3.3, “WIPLDEL”
- WIPUNDEL - see 3.3.4, “WIPUNDEL”
- WIPSHOW - see 3.3.5, “WIPSHOW”
- WIPCHANG - see 3.3.6, “WIPCHANG”
- WIPPARA - see 3.3.7, “WIPPARA”
- WIPCON - see 3.3.8, “WIPCON”
- WIPMERGE - see 3.3.9, “WIPMERGE”

WIPSHOW Example: To use the WIPSHOW command to display records for the Root and Derivation 2 files, you would type:

```
WIPSHOW  R  2
```

WIPLDEL Example with Multiple Files: When you specify more than one file in a command, the command executes against lines that include identifiers for any or all of the specified files. When you issue the WIPLDEL command below, the WIPLDEL command logically deletes lines that were inserted from the Derivation 1 file (%I-1), the Derivation 2 file (%I-2), or both files (% I-1,2):

```
WIPLDEL  1  2
```

WIPLDEL Example with a Single File: When you specify one file in a command, the command executes against lines that include the specified identifier. When you issue the command:

```
WIPLDEL  1
```

The WIPLDEL command logically deletes lines that were inserted from the Derivation 1 file (%I-1) or both files (% I-1,2).

3.3.2 WIPCOUNT

Purpose: The WIPCOUNT command counts the number of lines in the WIP member from the included Root, Derivation 1, and Derivation 2 files. The command places the result in a message line at the top of the WIP member. The message looks like this:

```
WIPCOUNT RESULTS:
NUMBER OF ROOT/COMMON LINES:
LINES INSERTED BY DERIVATION 1:
LINES INSERTED BY DERIVATION 2:
LINES INSERTED BY BOTH DERIVATION 1/2:
TOTAL COMMON AND INSERTED LINES:
```

Execution: To execute the WIPCOUNT command, type **WIPCOUNT** in the COMMAND field, then press ENTER.

Usage: Use the WIPCOUNT command to assess the condition of the WIP file.

Note: The WIPCOUNT command does not update the WIP header.

3.3.3 WIPLDEL

Purpose: The WIPLDEL command logically deletes all lines in the specified file or files, by placing a comment indicator (*) in the first column of the line. The command then automatically invokes WIPCOUNT to recount the WIP statistics.

Execution: To execute the WIPLDEL command, type **WIPLDEL** and the desired parameters (R, 1, 2) in the COMMAND field, then press ENTER.

Usage: Use the WIPLDEL command to:

- Compare any two of the three input files.
- View a single input member.

3.3.4 WIPUNDEL

Purpose: The WIPUNDEL command undeletes any lines logically deleted by WIPLDEL. The command then automatically invokes WIPCOUNT to recount the WIP statistics.

Execution: To execute the WIPUNDEL command, type **WIPUNDEL** and the desired parameters (R, 1, 2) in the COMMAND field, then press ENTER.

Usage: Use the WIPUNDEL in conjunction with the WIPLDEL and WIPSHOW commands to try out different editing scenarios before beginning the merge process.

3.3.5 WIPSHOW

Purpose: The WIPSHOW command displays the lines in either the Root, the Derivation 1, or the Derivation 2 file. It does not display lines that have been logically deleted.

Execution: The user first types the command **WIPSHOW R 1** in the COMMAND line. Below is an example of how the WIPSHOW command works.

```
EDIT ---- ENDEVOR.PDM.WIPLIB(WIPAA) - 01.01 ------------------ COLUMNS 001 072
COMMAND ===> WIPSHOW R 1                                       SCROLL ===> HALF
000009 ******* ----+----1----+----2----+----3----+----4----+----5----+----6----
000010         THIS IS ROOT LINE 1
000011 * I-1   INSERT BY DV1, LINE 1
000012 % I-1   INSERT BY DV1, LINE 2
000013 % I-2   INSERT BY DV2, LINE 1
000014         THIS IS A ROOT LINE
000015 % I-1,2 INSERT COMMON, LINE 1
000016 % I-1   INSERT BY DV1, LINE 3
000017         THIS IS A ROOT LINE
000018 % D-1,2 THIS IS A COMMON DELETE
000019 % I-2   INSERT BY DV2, LINE 1
000020 % I-1,2 INSERT BY DV2, LINE 1
000021         THIS IS A ROOT LINE
000022         THIS IS A ROOT LINE
000023 % I-2   INSERT BY DV2, LINE 2
000024 % I-1,2 INSERT COMMON, LINE 1
000025 % I-2   INSERT BY DV2, LINE 3
000026 % D-2   INSERT BY DV2, LINE 3
000027         THIS IS ROOT LINE 2
****** ***************************** BOTTOM OF DATA ****************************
```

Pressing ENTER executes the WIPSHOW command, which produces the following output.

```
EDIT ---- ENDEVOR.PDM.WIPLIB(WIPAA) - 01.02 ------------------ 11 LINES FOUND
COMMAND ===>                                                   SCROLL ===> HALF
****** ***************************** TOP OF DATA ******************************
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  9 LINE(S) NOT DISPLAYED
000010         THIS IS ROOT LINE 1
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  1 LINE(S) NOT DISPLAYED
000012 % I-1   INSERT BY DV1, LINE 2
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  1 LINE(S) NOT DISPLAYED
000014         THIS IS A ROOT LINE
000015 % I-1,2 INSERT COMMON, LINE 1
000016 % I-1   INSERT BY DV1, LINE 3
000017         THIS IS A ROOT LINE
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  2 LINE(S) NOT DISPLAYED
000020 % I-1,2 INSERT BY DV2, LINE 1
000021         THIS IS A ROOT LINE
000022         THIS IS A ROOT LINE
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  1 LINE(S) NOT DISPLAYED
000024 % I-1,2 INSERT COMMON, LINE 1
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  2 LINE(S) NOT DISPLAYED
000027         THIS IS ROOT LINE 2
****** ***************************** BOTTOM OF DATA ****************************
```

In this example, the WIPSHOW command displays only those lines that are from the Root and the Derivation 1 file and that have not been logically deleted. Line 11 is excluded from the display because it was logically deleted from the WIP member. Lines 15, 20 and 24 are included in the display because they are common between the Derivation 1 and the Derivation 2 file.

3.3.6 WIPCHANG

Purpose: The WIPCHANG command displays only the lines in the WIP file that have changed.

Execution: To execute the WIPCHANG command, type **WIPCHANG** in the COMMAND field, then press ENTER.

Usage: Use the WIPCHANG command to assess the condition of the WIP file.

3.3.7 WIPPARA

Purpose: The WIPPARA command identifies each paragraph in a WIP file. A paragraph is defined as a block of lines that have the same operation (insert or delete) from the same derivation file(s).

The WIPPARA command identifies each paragraph in the WIP file and adds a =NOTE= line immediately before the first line in the paragraph. The =NOTE= line identifies the type of paragraph. If the WIP file contains at least one paragraph, PDM positions the WIP member at the first paragraph found. To locate subsequent paragraphs in the WIP file, you can use the ISPF/PDF LOCATE SPECIAL command. The ISPF/PDF RESET command will remove the =NOTE= lines.

Execution: To execute the WIPPARA command, type **WIPPARA** in the COMMAND field then press ENTER.

Shown below is an example of the output from the WIPPARA command.

```
EDIT ---- ENDEVOR.PDM.WIPLIB(WA) - 01.00 ------------------- 6 PARAGRAPHS FOUND
COMMAND ===>                                                 SCROLL ===> HALF
=NOTE= * THE FOLLOWING PARAGRAPH IS A COMMON INSERT BLOCK
000019 % I-1,2 INSERT BY DV1, LINE 1
000020 % I-1,2 INSERT BY DV1, LINE 2
000021         THIS IS ROOT LINE 3
000022         THIS IS ROOT LINE 4
000023         THIS IS ROOT LINE 5
=NOTE= * THE FOLLOWING PARAGRAPH IS A CONFLICTING INSERT FROM DERIVATION 1
000024 %?I-1   INSERT BY DV1, LINE 3
000025 %?I-1   INSERT BY DV1, LINE 4
000026 %?I-1   INSERT BY DV1, LINE 5
=NOTE= * THE FOLLOWING PARAGRAPH IS A CONFLICTING INSERT FROM DERIVATION 2
000027 %?I-2   INSERT BY DV2, LINE 3
000028 %?I-2   INSERT BY DV2, LINE 4
000029 %?I-2   INSERT BY DV2, LINE 5
000030         THIS IS ROOT LINE 6
000031         THIS IS ROOT LINE 7
=NOTE= * THE FOLLOWING PARAGRAPH IS AN INSERT BLOCK FROM DERIVATION 2
000032 % I-2   INSERT BY DV2, LINE 6
000033 % I-2   INSERT BY DV2, LINE 7
=NOTE= * THE FOLLOWING PARAGRAPH IS A COMMON DELETE BLOCK
000034 % D-1,2 THIS IS ROOT LINE 8
```

Usage: Use the WIPPARA command to assess the condition of the WIP file.

3.3.8 WIPCON

Purpose: The WIPCON command displays only those lines identified as conflicts within the WIP member.

Execution: To execute the WIPCON command, type **WIPCON** in the COMMAND field, then press ENTER.

Usage: Use the WIPCON command to view quickly those areas of the WIP file where you will need to resolve conflicts.

3.3.9 WIPMERGE

Purpose: The WIPMERGE command allows you to test edits by creating a temporary merge file within an edit session. If you do not like the results of the temporary merge file, you can return to the WIP file and continue editing it.

The WIPMERGE command:

- Executes pending edit commands.
- Saves the current WIP member.
- Displays the merge file in browse mode.

When you end the browse session, the WIP edit session resumes.

Execution: To execute the WIPMERGE command, type **WIPMERGE** and, optionally, a single character change identifier in the COMMAND field, then press ENTER.

Usage: WIPMERGE uses the single optional character in the command line to identify inserted lines in the merge file. PDM places the identifier in column 1 of the merge file when a line has been inserted by either the Derivation 1 or Derivation 2 file.

Example: The command:

```
WIPMERGE  C
```

Instructs PDM to put the letter C in the first column of any lines in the merge file that were inserted by the Derivation 1 or Derivation 2 files.

When you specify a change ID in the WIPMERGE command, the message **Change ID Specified** appears in the upper right hand corner of the merge file.

3.4 The Next Step

When you have finished editing the WIP File, press PF3; the Edit - Entry Panel reappears. The message **Member Saved** appears in the upper right-hand corner of the screen.

```
Edit WIP -------------------- Edit - Entry Panel ----------------MEMBER SAVED
  Command ===>

  ISPF Library:
     Project ===>  ENDEVOR
     Group   ===>  PDM
     Type    ===>  WIPLIB
     Member  ===>  PROGX           (Blank or pattern for member selection list)

  Other partitioned or sequential data set:
     Data Set Name  ===>
```

This message indicates the edited WIP member is saved. Press PF3 to return to the AllFusion Endevor Change Manager Parallel Development Option menu.

Editing the WIP File is the second step in the PDM process. The next step is merging the edited WIP File into an output source file, which can then be added back into Endevor, stored in the appropriate source repository, or compiled.

Chapter 4. Merging WIP Files

4.1 The Merge Output File

PDM creates a merge output file from a WIP file. Before creating a merge output file, all conflicts must be edited out of the WIP file. The WIP file can contain non-conflicting inserts and deletes.

When building a WIP file, PDM:

- Removes the WIP header information.
- Inserts and deletes lines as necessary.
- Removes the WIP annotations.

A portion of a WIP file and the resulting merge output file is shown below.

4.1.1 WIP File

```
ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT REPORT-FILE ASSIGN U-T-SYSOUT.
    SELECT INPUT-FILE ASSIGN U-T-INPUT.
DATA DIVISION.
FILE SECTION.
FD REPORT-FILE
    LABEL RECORDS ARE OMITTED
    RECORDING MODE IS F
    RECORD CONTAINS 133 CHARACTERS
    DATA RECORD IS OUTPUT-RECORD.
  01 OUTPUT-RECORD.
    02 CARRIAGE-CONTROL PIC X.
    02 OUTPUT-LINE      PIC X(132).
FD INPUT-FILE
    LABEL RECORDS ARE STANDARD
    RECORD CONTAINS 80 CHARACTERS
    DATA RECORD IS INPUT-FILE-RECORD.
  01 INPUT-FILE-RECORD.
    02 INPUT-FIELD1           PIC 9(8).
    02 INPUT-FIELD2           PIC X(8).
    02 INPUT-FIELD3      PIC X(20).
    02 INPUT-FIELD4           PIC 9(6).
    02 INPUT-FIELD5           PIC 9(6).
    02 FILLER                 PIC X(32).
```

4.1.2 Merge Output File Based on Edits to WIP File

```
               ENVIRONMENT DIVISION.
               INPUT-OUTPUT SECTION.
               FILE-CONTROL.
                   SELECT REPORT-FILE ASSIGN U-T-SYSOUT.
                   SELECT INPUT-FILE ASSIGN U-T-INPUT.
               DATA DIVISION.
               FILE SECTION.
               FD REPORT-FILE
                   LABEL RECORDS ARE OMITTED
                   RECORDING MODE IS F
                   RECORD CONTAINS 133 CHARACTERS
                   DATA RECORD IS OUTPUT-RECORD.
                 01 OUTPUT-RECORD.
                   02 CARRIAGE-CONTROL PIC X.
                   02 OUTPUT-LINE      PIC X(132).
               FD INPUT-FILE
                   LABEL RECORDS ARE STANDARD
                   RECORD CONTAINS 80 CHARACTERS
                   DATA RECORD IS INPUT-FILE-RECORD.
                 01 INPUT-FILE-RECORD.
                   02 INPUT-FIELD1          PIC 9(8).
%?I-1              02 INPUT-FIELD2          PIC X(8).
%?I-2              02 INPUT-FIELD2          PIC X(6).
% D-1,2            02 INPUT-FIELD2          PIC X(10).
                   02 INPUT-FIELD3          PIC X(20).
                   02 INPUT-FIELD4          PIC 9(6).
                   02 INPUT-FIELD5          PIC 9(6).
%?I-1              02 FILLER                PIC X(32).
%?I-2              02 FILLER                PIC X(34).
```

In this example, two steps were involved in creating the merge output file:

1. The user deleted from the WIP file both inserts from Derivation 2, namely:

```
%?I-2              02 INPUT-FIELD2          PIC X(6).
%?I-2              02 FILLER                PIC X(34).
```

2. During the merge process, PDM:
   - Inserted the two Derivation 1 inserts. The lines annotated with %?I-1.
   - Made the common delete. The line annotated with % D-1,2.
   - Removed the annotation entries to produce the merge output file.

4.2 Creating a Merge Output File

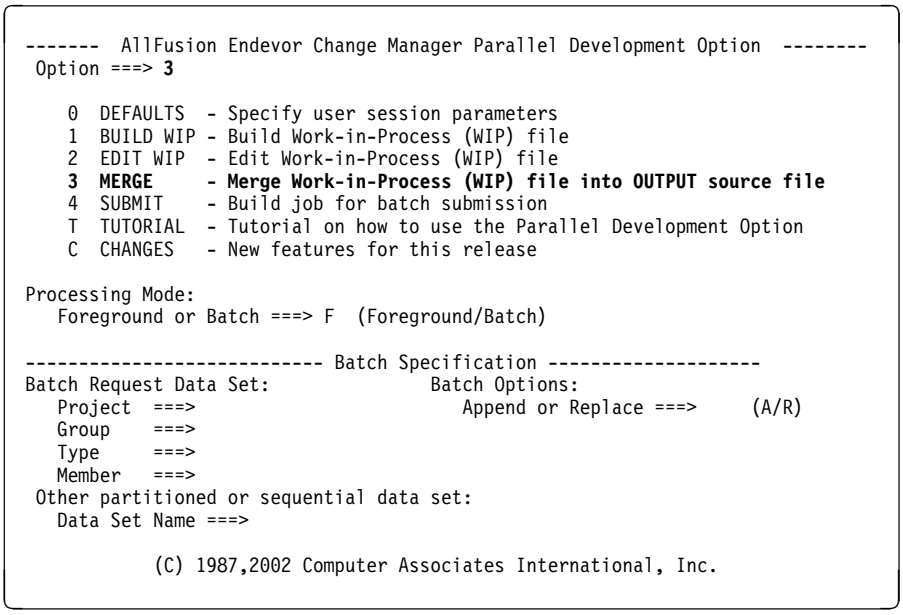
PDM creates a merge output file:

- Automatically, during the build WIP process, when you specify AUTOMATICALLY MERGE=Y on the WIP Specification Panel and there are no conflicts in the WIP file. This is described in Chapter 2, "Building WIP Files."
- Upon request, by invoking option 3 on the AllFusion Endevor Change Manager Parallel Development Option menu This procedure is described below.

4.2.1 Procedure: Merging a WIP File

To merge a WIP File, do the following:

1. Type **3** in the Command field on the AllFusion Endevor Change Manager Parallel Development Option menu. and specify the appropriate processing mode in the PROCESSING MODE: FOREGROUND OR BATCH field:
   - F — Foreground
   - B — Background



   Press ENTER to display the Merge Work In Process File screen.

2. Define the WIP file that you want to merge, using standard ISPF conventions.

```
Merge ------------------  Merge Work In Process File  -------------------------
 Command ===>
 Specify Work in Process ("WIP") Library below, then Press ENTER Key
 WIP Library:
    Project ===> endevor
    Library ===> pdm
    Type    ===> wiplib
    Member  ===> progx             (Blank or pattern for member selection list,
                                    '*' for all members)
 WIP Other partitioned or sequential data set:
    Data Set Name ===>
```

When specifying a WIP member on this panel, you can:

- Enter an explicit member name, to merge only that member.
- Enter an asterisk (*) to merge all members in the library.
- Leave the MEMBER field blank or enter a name mask to select a member(s) from the Merge Output Member Selection List.

When you press ENTER after typing in the appropriate information, the Merge Output Specification Panel is returned.

3. Specify the library and member to which you want PDM to write the merge output file, using standard ISPF conventions. Press ENTER when you have typed in all pertinent data.

```
Merge ---------------- Merge Output Specification Panel -----------------------
 Command ===>
 Specify MERGE OUTPUT location below, then press ENTER key.
 Merge Library:
    Project ===>
    Group   ===>
    Type    ===>
    Member  ===>                 (Pattern for member name overlay)
 Merge Other partitioned or sequential data set:
    Data Set Name  ===>
 Merge Data Set Options:
    If partitioned, replace like-name members ===>      (Yes/No)
 Merge Options:
    Delete WIP member after MERGE ===>      (Yes/No)
    Generate MERGE reports        ===>      (All/Summary/No)
    Panvalet/Librarian language   ===>
    Write stats to data set ===>
```

When specifying the target member name, you can:

- Enter an explicit member name, to create a merged member in the output data set.
- Leave the field blank to select member names from the Merge Output Selection List.
- Provide an overlay mask to be applied to all member names.

  See the 4.3.1.1, “The Merge Output Member Selection List” on page 4-9 for more information about leaving the MEMBER field blank or assigning an overlay mask.

See 4.3, “The Merge Output Specification Panel” on page 4-8 for field descriptions.

4. One of the following occurs:

   - If you **specified an explicit member name or entered an asterisk** for all members, the Merge process is immediately invoked. When processing is complete, the Merge Work In Process File screen is returned with the message **MERGE COMPLETE** in the upper right corner.

```
Merge ------------------  Merge Work In Process File  -------- MERGE COMPLETE
 Command ===>
 Specify Work in Process ("WIP") Library below, then Press ENTER Key
 WIP Library:
    Project ===> endevor
    Library ===> pdm
    Type    ===> wiplib
    Member  ===> progx         (Blank or pattern for member selection list,
                                     '*' for all members)
 WIP Other partitioned or sequential data set:
    Data Set Name ===>
```

     This message indicates that the WIP File(s) specified on the screen was successfully merged into the output source file specified on the Merge Output Specification Panel.

   - If you **left the MEMBER field blank or provided a name mask**, you must select one or more members from the Merge Output Member Selection List, described in 4.3.1.1, “The Merge Output Member Selection List” on page 4-9. You complete the merge process from this selection list.

4.3 The Merge Output Specification Panel

```
Merge ---------------- Merge Output Specification Panel -----------------------
 Command ===>
 Specify MERGE OUTPUT location below, then press ENTER key.
 Merge Library:
    Project ===>
    Group   ===>
    Type    ===>
    Member  ===>                 (Pattern for member name overlay)
 Merge Other partitioned or sequential data set:
    Data Set Name  ===>
 Merge Data Set Options:
    If partitioned, replace like-name members ===> Y    (Yes/No)
 Merge Options:
    Delete WIP member after MERGE ===> N    (Yes/No)
    Generate MERGE reports        ===> A    (All/Summary/No)
    Panvalet/Librarian language   ===>
    Write stats to data set ===>
```

4.3.1 Merge Output Specification Panel Fields

Field	Description
Merge Library fields **1**	Used to specify the merge library, using standard ISPF conventions.
Merge Other Partitioned or Sequential Data Set	Used as an alternative for specifying the merge library and target member. Specify using standard ISPF conventions.
If Partitioned, Replace Like-Named Members **2**	Indicates if you want to replace like-named members in the data set. Valid values are: ■ Y — Yes ■ N — No
Delete WIP Member After Merge	Required. Specifies if the WIP member should be deleted after the merge action. Valid values are: ■ Yes — PDM deletes the WIP member if there are no errors on the MERGE action and the WIP data set is a PDS. ■ No — Default. Do not delete the WIP member after a MERGE action.
Generate MERGE Reports	Required. Acceptable values are: ■ A — Default. Generates both the Merge Summary and Detail Reports. ■ S — Generates only the Merge Summary Report. ■ N — Prevents the generation of any reports.

Field	Description
Panvalet/ Librarian Language **3**	Optional. This field associates an AllFusion CA-Panvalet or AllFusion CA-Librarian type identifier with the named output member. Selections include: ■ AllFusion CA-Panvalet: ALC, ANSCOBOL, AUTOCODE, BAL, COBOL, COBOL-72, DATA, FORTRAN, JCL, OBJECT, PL/1, RPG, USER180, USER780, OTHER. If left blank, the default value is DATA. ■ AllFusion CA-Librarian: ASM, COB, DAT, FOR, FRG, FRS, GIF, GOF, JCL, PLF, PLI, RPG, TXT, VSB. If left blank, the default value is DAT.
Write Stats to Data Set	Optional. Specifies the name of the data set used to capture PDM statistics. If specified, this field must conform to the normal ISPF/PDF naming conventions. See Appendix D, "Statistical Data Control Blocks" information about the characteristics of the statistics data set.

Note:
1 : The record length of the Merge data set must be large enough to support the largest WIP record. Generally this means that the Merge data set must be no shorter than eight bytes less than the WIP record length. For example, if the WIP data set record length is 88, then the Merge data set record length must be at least 80.

2 : This option does not apply to sequential files.

3 : PDM ignores the field if the merge data set is not an AllFusion CA-Panvalet or AllFusion CA-Librarian data set.

4.3.1.1 The Merge Output Member Selection List

PDM produces the Merge Output Member Selection List when you leave the MEMBER field blank or provide a name mask on the Merge Output Specification panel. PDM builds this list using the members in the WIP library.

```
Merge ---------------- Merge Output Member Selection List ---------------------
  Command ===>                                                  Scroll ===>
    WIP Data Set Name:
     Output Data Set Name:
      Name      Rename     VV.MM  Created     Changed       Size   Init    ID
    BC1PAL00              01.00 01/07/25  01/07/25 18:22     66     66 ZSXPTB1
  s BC1PAL10              01.00 01/07/25  01/07/25 18:22     66     66 ZSXPTB1
    BC1PBDWK              01.00 01/07/25  01/07/25 16:30     17     17 ZSXPTB1
    BC1PBM30              01.00 01/07/25  01/07/25 16:30     17     17 ZSXPTB1
    BC1PBM35              01.00 01/07/25  01/07/25 18:00     15     15 ZSXPTB1
    BC1PBM40              01.00 01/07/25  01/07/25 20:52     66     66 ZSXLGB1
    BC1PBR10              01.00 01/07/25  01/07/25 20:52     66     66 ZSXLGB1
  s BC1PCAF               01.00 01/07/26  01/07/26 16:31     27     27 ZSXPTB1
    BC1PCIOB              01.00 01/07/26  01/07/26 16:25     55     55 ZSXPTB1
    BC1PCONG              01.00 01/07/25  01/07/25 16:30     17     17 ZSXPTB1
    BC1PCONP              01.00 01/07/25  01/07/25 16:30     15     15 ZSXPTB1
    BC1PC1PR              01.00 01/07/25  01/07/25 18:37     30     30 ZSXPTB1
    BC1PDCON              01.00 01/07/25  01/07/25 16:30     30     30 ZSXPTB1
    BC1PDSIN              01.00 01/07/25  01/07/25 18:21     66     66 ZSXPTB1
    BC1PEFLT              01.00 01/07/25  01/07/25 18:21     66     66 ZSXPTB1
    BC1PESSI              01.00 01/07/27  01/07/27 12:20     23     23 ZSXPTB1Q
    BC1PFOOT              01.00 01/07/27  01/07/27 12:24     17     17 ZSXPTB1Q
```

To select a member type an **S** next to the member name. When you press ENTER, the Merge process is invoked for that member and, upon completion, the message ***MERGED** appears in the RENAME field next to each member. Multiple WIP file members can be selected at one time.

```
Merge ---------------- Merge Output Member Selection List ---------------------
  Command ===>                                                  Scroll ===>
    WIP Data Set Name:
     Output Data Set Name:
      Name      Rename     VV.MM  Created     Changed       Size   Init    ID
    BC1PAL00              01.00 01/07/25  01/07/25 18:22     66     66 ZSXPTB1
    BC1PAL10  *MERGED     01.00 01/07/25  01/07/25 18:22     66     66 ZSXPTB1
    BC1PBDWK              01.00 01/07/25  01/07/25 16:30     17     17 ZSXPTB1
    BC1PBM30              01.00 01/07/25  01/07/25 16:30     17     17 ZSXPTB1
    BC1PBM35              01.00 01/07/25  01/07/25 18:00     15     15 ZSXPTB1
    BC1PBM40              01.00 01/07/25  01/07/25 20:52     66     66 ZSXLGB1
    BC1PBR10              01.00 01/07/25  01/07/25 20:52     66     66 ZSXLGB1
    BC1PCAF   *MERGED     01.00 01/07/26  01/07/26 16:31     27     27 ZSXPTB1
    BC1PCIOB              01.00 01/07/26  01/07/26 16:25     55     55 ZSXPTB1
    BC1PCONG              01.00 01/07/25  01/07/25 16:30     17     17 ZSXPTB1
    BC1PCONP              01.00 01/07/25  01/07/25 16:30     15     15 ZSXPTB1
    BC1PC1PR              01.00 01/07/25  01/07/25 18:37     30     30 ZSXPTB1
    BC1PDCON              01.00 01/07/25  01/07/25 16:30     30     30 ZSXPTB1
    BC1PDSIN              01.00 01/07/25  01/07/25 18:21     66     66 ZSXPTB1
    BC1PEFLT              01.00 01/07/25  01/07/25 18:21     66     66 ZSXPTB1
    BC1PESSI              01.00 01/07/27  01/07/27 12:20     23     23 ZSXPTB1Q
    BC1PFOOT              01.00 01/07/27  01/07/27 12:24     17     17 ZSXPTB1Q
```

Note: If you are using batch mode, the message ***WRITTEN** appears rather than *MERGED, indicating that the appropriate batch request control statements have been written. The actual merge does not take place until the requests are submitted for processing.

4.3.1.2 Overlay Masks and the Selection List

When you provide an overlay mask in the MEMBER field, PDM renames each selected WIP member in the output data set. The names that will be used in the merge output library appear in the RENAME field.

For example, if you specify an overlay mask of **W*** on the Merge Output Specification Panel, the Merge Output Member Selection List would appear as illustrated below:

```
Merge ---------------- Merge Output Member Selection List ---------------------
  Command ===>                                                 Scroll ===>
    WIP Data Set Name:
     Output Data Set Name:
      Name      Rename     VV.MM  Created     Changed       Size   Init    ID
    BC1PAL00   WC1PAL00   01.00 01/07/25  01/07/25 18:22     66     66 ZSXPTB1
    BC1PAL10   WC1PAL10   01.00 01/07/25  01/07/25 18:22     66     66 ZSXPTB1
    BC1PBDWK   WC1PBDWK   01.00 01/07/25  01/07/25 16:30     17     17 ZSXPTB1
    BC1PBM30   WC1PBM30   01.00 01/07/25  01/07/25 16:30     17     17 ZSXPTB1
    BC1PBM35   WC1PBM35   01.00 01/07/25  01/07/25 18:00     15     15 ZSXPTB1
    BC1PBM40   WC1PBM40   01.00 01/07/25  01/07/25 20:52     66     66 ZSXLGB1
    BC1PBR10   WC1PBR10   01.00 01/07/25  01/07/25 20:52     66     66 ZSXLGB1
    BC1PCAF    WC1PCAF    01.00 01/07/26  01/07/26 16:31     27     27 ZSXPTB1
    BC1PCIOB   WC1PCIOB   01.00 01/07/26  01/07/26 16:25     55     55 ZSXPTB1
    BC1PCONG   WC1PCONG   01.00 01/07/25  01/07/25 16:30     17     17 ZSXPTB1
    BC1PCONP   WC1PCONP   01.00 01/07/25  01/07/25 16:30     15     15 ZSXPTB1
    BC1PC1PR   WC1PC1PR   01.00 01/07/25  01/07/25 18:37     30     30 ZSXPTB1
    BC1PDCON   WC1PDCON   01.00 01/07/25  01/07/25 16:30     30     30 ZSXPTB1
    BC1PDSIN   WC1PDSIN   01.00 01/07/25  01/07/25 18:21     66     66 ZSXPTB1
    BC1PEFLT   WC1PEFLT   01.00 01/07/25  01/07/25 18:21     66     66 ZSXPTB1
    BC1PESSI   WC1PESSI   01.00 01/07/27  01/07/27 12:20     23     23 ZSXPTB1Q
    BC1PFOOT   WC1PFOOT   01.00 01/07/27  01/07/27 12:24     17     17 ZSXPTB1Q
```

When you have finished selecting members for the output source file, press PF3. The Merge Work In Process File screen is returned. Because you used the selection list to perform the Merge process, no message appears in the upper right corner of the screen.

4.4 The Next Step

The next step in the PDM process depends upon whether you are using batch or foreground mode. If you are using foreground, the Merge process is performed immediately. If you are using batch, you must submit for processing the Merge requests you have just generated. This procedure is described in the next chapter.

Three reports are produced when the Merge step is processed in batch mode. These reports, which are listed below, are discussed in detail in Chapter 6, "PDM Reports":

- PDM Merge Output Detail Report
- PDM Merge Output Summary Report
- PDM Syntax Request Report

Review these reports as necessary.

Once you have created a merge output file, you can:

- Add the file back into Endevor, using the Add/Update capability.
- Store the file in the appropriate source repository.
- Input the file to a compiler.

Chapter 5. Using PDM in Batch

5.1 Generating Build WIP and Merge Requests

You can generate Build WIP and Merge requests for batch submission by:

- Using the appropriate options from the AllFusion Endevor Change Manager Parallel Development Option menu: option **1** for Build WIP requests, option **3** for Merge requests.
- Coding your own job. See Appendix B, "Batch Syntax" for PDM batch syntax.

You can then submit these requests for execution by:

- Using option **4**, SUBMIT, from the AllFusion Endevor Change Manager Parallel Development Option menu.
- Submitting your own JCL. Refer to Appendix A, "Batch Execution JCL" for a sample of the required batch execution JCL.

5.2 Specifying a Request Data Set

The first step in generating a PDM batch request is to specify a data set in which to store the requests. You do this in the batch specification fields on the AllFusion Endevor Change Manager Parallel Development Option menu.

```
-------  AllFusion Endevor Change Manager Parallel Development Option  --------
 Option ===>

    0  DEFAULTS  - Specify user session parameters
    1  BUILD WIP - Build Work-in-Process (WIP) file
    2  EDIT WIP  - Edit Work-in-Process (WIP) file
    3  MERGE     - Merge Work-in-Process (WIP) file into OUTPUT source file
    4  SUBMIT    - Build job for batch submission
    T  TUTORIAL  - Tutorial on how to use the Parallel Development Option
    C  CHANGES   - New features for this release

Processing Mode:
   Foreground or Batch ===> b  (Foreground/Batch)

---------------------------- Batch Specification --------------------
Batch Request Data Set:               Batch Options:
   Project  ===> endevor                 Append or Replace ===> r   (A/R)
   Group    ===> ndvr
   Type     ===> batch
   Member   ===> reqcards
   Data Set Name ===>

            (C) 1987,2002 Computer Associates International, Inc.
```

5.2.1 Batch Request Data Set Panel Field Descriptions

Field	Description
Batch Request Data Set fields	The batch request data set must be either a sequential data set or a partitioned data set with an explicitly identified member. The data set can contain either: ■ Fixed length records — LRECL must be 80 ■ Variable length records — LRECL must be 84
Other Partitioned or Sequential Data Set	Used as an alternative for specifying a request data set, using standard ISPF conventions.
Append or Replace	Required. Valid values are: ■ R — Default. PDM replaces like-named members in the request data set ■ A — PDM appends the statements to an existing data set.

5.2.2 Request Dataset Allocation Panel

If the request data set you specify is not allocated, PDM displays a Request Dataset Allocation panel when you press ENTER.

```
PDM ------------------- Request Dataset Allocation Pan   Dataset not Catalogued
 Command ===>
 Dataset Name: ENDEVOR.JSMITH.WORKLIB
   Management Class     ===>             (Blank for default management class)
   Storage Class        ===>             (Blank for default storage class)
     Volume Serial      ===>             (Blank for authorized default volume)
   Data Class           ===>             (Blank for default data class)
     Space Units        ===>             (BLKS, TRKS, CYLS)
     Primary Quantity   ===>             (In above units)
     Secondary Quantity ===>             (In above units)
     Directory Blocks   ===>             (Zero for sequential data set)
     Record Format      ===> FB          (F, FB, V, VB)
     Record Length      ===> 80
     Block Size         ===> 14520       (Blank for system determined value;
                                          Required when Space Units is BLKS)
   *Note: The Batch Request dataset record length must be 80 for fixed
    length records and 84 for variable length records.
```

The Request Dataset Allocation Panel allows you to allocate the batch AllFusion Endevor Change Manager Parallel Development Option menu.

5.2.2.1 Request Dataset Allocation Panel Field Descriptions

Field	Description
Management Class	Optional. If you specify a management class that is not defined, the allocation fails and PDM issues an error message.
Storage Class	Optional. If you specify a storage class that is not defined, the allocation fails and PDM issues an error message.
Volume Serial	Optional. If you specify a volume serial that is not defined, the allocation fails with a recoverable application error.
Data Class	Optional. If you specify a data class that is not defined, the allocation fails and PDM issues an error message.
Space Units	Acceptable values are: ■ BLKS — Blocks ■ TRKS — Tracks ■ CYLS — Cylinders
Primary Quantity	Must be numeric, and in the same units as the SPACE UNITS field.
Secondary Quantity	Must be numeric, and in the same units as the SPACE UNITS field.

Field	Description
Directory Blocks	Must be numeric. If you specify a request member, you must provide a non-zero value in this field. If you want to allocate a sequential data set, this value must be zero.
Record Format	Acceptable values are: ■ F ■ FB ■ V ■ VB
Record Length	Must be numeric, 80 for fixed length records and 84 for variable length records.
Block Size	Required if the space unit is BLKS (blocks), otherwise optional. If left blank, block size is set to zero. If provided, the value must be numeric and greater than zero. If the block size is specified, if it is not zero, and the record format is: ■ F or FB — The block size must be an integral multiple of the record length. ■ V or VB — The block size must be at least four bytes larger than the record length.

5.3 Generating Build WIP Requests Option 1

To generate Build WIP request statements for later job submission, select option **1** on the AllFusion Endevor Change Manager Parallel Development Option menu. Make sure that you:

- Indicate **B** for batch processing in the PROCESSING MODE: FOREGROUND OR BATCH field.
- Provide the required BATCH SPECIFICATION information. See the previous section if you need to review this information.

```
-------  AllFusion Endevor Change Manager Parallel Development Option  --------
 Option ===> 1

    0  DEFAULTS  - Specify user session parameters
    1  BUILD WIP - Build Work-in-Process (WIP) file
    2  EDIT WIP  - Edit Work-in-Process (WIP) file
    3  MERGE     - Merge Work-in-Process (WIP) file into OUTPUT source file
    4  SUBMIT    - Build job for batch submission
    T  TUTORIAL  - Tutorial on how to use the Parallel Development Option
    C  CHANGES   - New features for this release

Processing Mode:
   Foreground or Batch ===> b  (Foreground/Batch)

---------------------------- Batch Specification --------------------
Batch Request Data Set:               Batch Options:
   Project  ===> endevor                 Append or Replace ===> r   (A/R)
   Group    ===> ndvr
   Type     ===> batch
   Member   ===> reqcards

            (C) 1987,2002 Computer Associates International, Inc.
```

Press ENTER. You receive the same series of screens that appear when you process PDM to build a WIP File in foreground mode. When you have filled in the last panel or issued the create command on the WIP Member Selection Matrix, the AllFusion Endevor Change Manager Parallel Development Option menu reappears, with the message **Requests Written** in the upper right-hand corner of the screen This message indicates that the Build WIP requests have been written to the request data set specified on the AllFusion Endevor Change Manager Parallel Development Option menu.

5.4 Generating Merge Requests Option 3

To generate Merge requests for later job submission, select option **3** on the AllFusion Endevor Change Manager Parallel Development Option menu. Make sure that you:

- Indicate **B** for batch processing in the PROCESSING MODE: FOREGROUND OR BATCH field.
- Provide the required BATCH SPECIFICATION information. See the first section in this chapter if you need to review this information.

```
-------  AllFusion Endevor Change Manager Parallel Development Option  --------
 Option ===> 3

    0  DEFAULTS  - Specify user session parameters
    1  BUILD WIP - Build Work-in-Process (WIP) file
    2  EDIT WIP  - Edit Work-in-Process (WIP) file
    3  MERGE     - Merge Work-in-Process (WIP) file into OUTPUT source file
    4  SUBMIT    - Build job for batch submission
    T  TUTORIAL  - Tutorial on how to use the Parallel Development Option
    C  CHANGES   - New features for this release

Processing Mode:
   Foreground or Batch ===> b  (Foreground/Batch)

---------------------------- Batch Specification --------------------
Batch Request Data Set:              Batch Options:
   Project  ===> endevor                Append or Replace ===> r   (A/R)
   Group    ===> ndvr
   Type     ===> batch
   Member   ===> reqcards
 Other partitioned or sequential data set:
   Data Set Name ===>

            (C) 1987,2002 Computer Associates International, Inc.
```

  Press ENTER. You will receive the same series of screens that appear when you process PDM to merge the WIP File in foreground mode. Follow the procedure to merge the file into an output source file. See Chapter 4, “Merging WIP Files” of this manual to review this procedure.

If you provide a MEMBER name in the Merge Output Specification Panel, the Merge Work In Process File screen is returned when you press ENTER. The message **Requests Written** appears in the upper right-hand corner, indicating that the batch request has been written.

If you leave the MEMBER field blank or specify a name mask on the Merge Output Specification panel, the Merge Output Member Selection List is returned when you press ENTER. Select the member(s) to be merged, then press ENTER. The selection list reappears with the message ***WRITTEN** in the RENAME field next to the selected members.

5.5 Submitting Batch Requests for Execution Option 4

To submit batch jobs, select option **4** from the AllFusion Endevor Change Manager Parallel Development Option menu. Make sure you:

- Indicate **B** for batch processing in the PROCESSING MODE: FOREGROUND OR BATCH field.
- Provide the required BATCH SPECIFICATION information. See the first section in this chapter if you need to review this information.

```
-------  AllFusion Endevor Change Manager Parallel Development Option  --------
 Option ===> 4

    0  DEFAULTS  - Specify user session parameters
    1  BUILD WIP - Build Work-in-Process (WIP) file
    2  EDIT WIP  - Edit Work-in-Process (WIP) file
    3  MERGE     - Merge Work-in-Process (WIP) file into OUTPUT source file
    4  SUBMIT    - Build job for batch submission
    T  TUTORIAL  - Tutorial on how to use the Parallel Development Option
    C  CHANGES   - New features for this release

Processing Mode:
   Foreground or Batch ===> b  (Foreground/Batch)

---------------------------- Batch Specification --------------------
Batch Request Data Set:              Batch Options:
   Project  ===> endevor                Append or Replace ===> r   (A/R)
   Group    ===> ndvr
   Type     ===> batch
   Member   ===> reqcards
```

Press ENTER to display the PDM Batch Options Menu.

5.5.1 The PDM Batch Options Menu

The PDM Batch Options Menu allows you to browse, edit (as necessary), and subsequently submit the request data set you have defined. You can also build additional JCL to be included with the request. Listed below is a brief explanation of each processing option available:

```
Submit --------------------- PDM Batch Options Menu --------------------------
  Option ===>
      1  BROWSE    - Browse request data set
      2  EDIT      - Edit request data set
      3  BUILD JCL - Enter additional JCL to be included with the job
      4  SUBMIT    - Submit job for batch processing
   Batch Request Data Set:                                      (Req'd for
      Project  ===> endevor         Include JCL ===> n   (Y/N)   option 4)
      Group    ===> ndvr
      Type     ===> batch
      Member   ===>            (Option 1 or 2: Blank for member selection list)
   Other partitioned or sequential data set:
   Data Set Name ===>
   JOB Statement Information:
    ===> //userid   JOB (00000000),'ENDEVOR',NOTIFY=user-name,
    ===> //*       REGION=region-value,MSGCLASS=msgclass,CLASS=job-class
    ===> //*
```

Use This Option	To
1	Browse the request data set to ensure that the information it contains is correct. The BROWSE option uses a standard ISPF/PDF Browse facility, and lists the request card syntax as it was generated.
2	Edit the request data set, if necessary. The EDIT option uses the standard ISPF/PDF Edit facility.
3	Define additional JCL (generally DD statements) to be included with the JCL to be submitted.
4	Submit a job that executes the requests in the request data set.

5.5.1.1 Browsing the Request Data Set

Select option **1** from the PDM Batch Options Menu to review the request data set specified. When you press ENTER, a standard TSO/ISPF Browse panel returns.

```
BROWSE -- ENDEVOR.NDVR.BATCH(REQCARDS) ----------------- LINE 00000000 COL 001 080
 COMMAND ===>                                             SCROLL ===> PAGE
******************************** TOP OF DATA  ********************************
BUILD WIP  DSNAME     'ENDEVOR.PDM.WIPLIB                          '
      ROOT DSNAME     'ENDEVOR.PDM.ROOTLIB                         '
      DV1  DSNAME     'ENDEVOR.PDM.DV1LIB                          '
      DV2  DSNAME     'ENDEVOR.PDM.DV2LIB                          '
           COMPARE    007 THRU 072
           REPLACE    .
WIP BC1PAL00   ROOT BC1PAL00
               DV1  BC1PAL00
               DV2  BC1PAL00
                                              .
WIP BC1PAL10   ROOT BC1PAL10
               DV1  ' '
               DV2  ' '
                                              .
WIP BC1PBDWK   ROOT BC1PBDWK
               DV1  ' '
               DV2  BC1PBDWK
                                              .
WIP BC1PBM30   ROOT BC1PBM30
               DV1  ' '
               DV2  ' '
```

Use the PF7 and PF8 keys to page through the request data set. When you are done browsing the data set, press PF3 to return to the PDM Batch Options Menu.

5.5.1.2 Editing the Request Data Set

Select option **2** from the PDM Batch Options Menu to edit the request data set specified. You can either change the existing Build WIP requests or add new requests using the PDM syntax (described in Appendix B, “Batch Syntax”). When you press ENTER, a standard ISPF/PDF Edit panel is returned.

```
EDIT ---- ENDEVOR.NDVR.BATCH(REQCARDS) - 01.00 ------------------- COLUMNS 001 072
 COMMAND ===>                                                 SCROLL ===> CSR
 ****** ***************************** TOP OF DATA ****************************
 000001 BUILD WIP  DSNAME     'ENDEVOR.PDM.WIPLIB                          '
 000002       ROOT DSNAME     'ENDEVOR.PDM.ROOTLIB                         '
 000003       DV1  DSNAME     'ENDEVOR.PDM.DV1LIB                          '
 000004       DV2  DSNAME     'ENDEVOR.PDM.DV2LIB                          '
 000005            COMPARE    007 THRU 072
 000006            REPLACE    .
 000007 WIP BC1PAL00   ROOT BC1PAL00
 000008                DV1  BC1PAL00
 000009                DV2  BC1PAL00
 000010                                              .
 000011 WIP BC1PAL10   ROOT BC1PAL10
 000012                DV1  ' '
 000013                DV2  ' '
 000014                                              .
 000015 WIP BC1PBDWK   ROOT BC1PBDWK
 000016                DV1  ' '
 000017                DV2  BC1PBDWK
 000018                                              .
 000019 WIP BC1PBM30   ROOT BC1PBM30
 000020                DV1  ' '
.000021                DV2  ' '
```

Edit the WIP request data set as necessary. Refer to Appendix B, “Batch Syntax” for additional information about coding batch requests. When you finish, press PF3 to return to the PDM Batch Options Menu.

5.5.1.3 Building JCL to be Included with the Batch Request

Select option **3** from the PDM Batch Options Menu to define JCL (usually DD statements) that should be included with the JCL submitted to execute the request data set. You might use this option if, for example, a request referenced a source or target file by DDNAME.

When you press ENTER, the following screen returns:

```
Submit ---------- JCL to be Included with the PDM Batch Job -------------------
  Command ===>
         .........1.........2.........3.........4.........5.........6.........7..
    ===>
    ===>
    ===>
    ===>
    ===>
    ===>
    ===>
    ===>
    ===>
    ===>
    ===>
    ===>
    ===>
    ===>
    ===>
    ===>
    ===>
    ===>
    ===>
    ===>
```

Enter the JCL, in complete statements, that you want to include with the batch job.

5.5.1.4 Submitting the Request Data Set for Batch Processing

Select option **4** to submit the request data set for execution. In addition to specifying the WIP data set, you must also indicate whether you want to include any additional JCL that was defined (by option **3** above).

```
Submit --------------------- PDM Batch Options Menu --------------------------
  Option ===>
      1  BROWSE    - Browse request data set
      2  EDIT      - Edit request data set
      3  BUILD JCL - Enter additional JCL to be included with the job
      4  SUBMIT    - Submit job for batch processing
   Batch Request Data Set:                                        (Req'd for
      Project  ===> endevor         Include JCL ===> n   (Y/N)   option 4)
      Group    ===> ndvr
      Type     ===> batch
      Member   ===>            (Option 1 or 2: Blank for member selection list)
   Other partitioned or sequential data set:
   Data Set Name ===>
   JOB Statement Information:
    ===> //userid   JOB (00000000),'ENDEVOR',NOTIFY=user-name,
    ===> //*       REGION=region-value,MSGCLASS=msgclass,CLASS=job-class
    ===> //*
```

Enter the following information, then press ENTER to submit the job.

Field	Description
Include JCL	Required. Indicates whether you want to include additional JCL when you submit the batch request. Valid values are: ■ Y — Yes, include the JCL. The JCL must be defined, using option **3**, BUILD JCL, before entering **Y** in this field.) ■ N — No. Default.
Job Statement Information	This is the job card to be submitted with the job.

A message similar to that listed on the screen below is returned to inform you that the job has been submitted:

```
*** IKJ56250I JOB JSMITH1F(JOB01743) SUBMITTED
```

Chapter 6. PDM Reports

6.1 General Information

When you process your Build WIP and Merge requests in batch mode, PDM produces several reports. Use the Build WIP reports to determine the complexity of a project. Use the Merge Output reports to review the outcomes of the Merge process.

The PDM reports contain the following information:

- The **PDM Syntax Request Report** (for Build WIP) lists the exact syntax you used to request the Build WIP operation.
- The **PDM Build WIP Detail Report** lists every member set processed and provides detail information for each member set. You can optionally suppress the detail information from printing.
- The **PDM Build WIP Summary Report** lists summary information about the entire Build action.
- The **PDM Syntax Request Report** (for Merge) lists the exact syntax you used to request the Merge operation.
- The **PDM Merge Output Detail Report** lists the WIP members used to create the merge output source file and detail information about each member. You can optionally suppress the detail information from printing.
- The PDM **Merge Output Summary Report lists** summary information about the Merge process.

Warning: If you code the STATS OFF clause using the DETAIL option, all reports are produced, but the Build WIP Detail and Merge Output Detail reports do not contain detail statistics for each member set. If you code the clause using the ALL option, however, only the Syntax reports are produced. The Detail and Summary reports are not produced at all.

In batch mode only, PDM writes the Syntax, Detail, and Summary Reports to the C1MSGS1 DD statement. It is possible to write the summary reports to a separate data set. If the C1MSGS2 DD statement is defined in the PDM execution JCL, PDM writes the Build Summary and Merge Summary Reports to the C1MSGS2 file. The C1MSGS2 DD statement can be added to the PDM batch execution JCL, the PDM batch execution skeleton, or you can specify it as additional, included JCL in the PDM Submit option. See Appendix A, “Batch Execution JCL” for information on modifying PDM JCL.

When running PDM in foreground, the action reports are written to a data set named *userid*.C1TEMPR_.MSGS. If you want to review the reports, split the ISPF screen and use the ISPF/PDF browse function to view the data set. This messages data set is automatically deleted when PDM terminates.

6.2 PDM Syntax Request Report for Build WIP

```
COPYRIGHT (C) Computer Associates, INC., 2002                                    24JAN02 09:52:47       PAGE   1
                                Parallel Development Option Syntax Request Report            RELEASE x.x  SERIAL xnnnn
         09:52:47  PDM0200I  PDM control statement parsing is beginning
          BUILD WIP  DSNAME     'ENDEVOR.PDM.WIPLIB'
                ROOT DSNAME     'ENDEVOR.NQAPROD2.SRCLIB'
                DV1  ENVIRONMENT 'ENDEVOR'
                     SYSTEM      'NDVR'
                     SUBSYSTEM   'PDM     '
                     TYPE        'ASMIPGMR'
                     STAGE NUMBER  2
                DV2  DSNAME     'ENDEVOR.PDM.SRCLIB'
                     COMPARE   001  THRU 072
                     REPLACE
                       .
            WIP  'BC1G2* '
            ROOT 'BC1G2* '
            DV1  'BC1G2* '
            DV2  'BC1G2* '
                       .
         09:52:47  PDM0201I  PDM control statement parsing has successfully completed
```

6.2.1 Syntax Validation

When you enter your Build WIP requests, the Parallel Development Option validates the syntax in the WIP request data set. The PDM Syntax Request Report mirrors exactly what you entered, in the order in which it was entered, and flags any syntax errors.

When all syntax requests have been checked, PDM looks for errors. If no errors can be found, processing continues. If errors do exist, processing terminates at this point. Refer to the *Error Codes and Messages Guide* for an explanation of any error messages received. Note the last line in the PDM Syntax Request Report above; this message indicates that all request cards have been processed successfully.

6.3 PDM Build WIP Detail Report

```
COPYRIGHT (C) Computer Associates, INC., 2002                                    24JAN02 09:53:00       PAGE   1
                        Parallel Development Option BUILD WIP Detail Report          RELEASE x.x  SERIAL Xnnnn
                    *---------------------------------------*
                    *         Beginning of BUILD WIP        *
                    *---------------------------------------*
                    WIP:            ENDEVOR.PDM.WIPLIB
                    Member:         BC1G2000
                    Root:           ENDEVOR.NQAPROD2.SRCLIB
                    Member:         BC1G2000
                    Derivation 1:   ENDEVOR /P/NDVR /PDM     /ASMIPGMR
                    Member:         BC1G2000   01.31
                    Derivation 2:   ENDEVOR.PDM.SRCLIB
                    Member:         BC1G2000
        09:53:03  PDM0064I  BUILD WIP operation is complete.  Return Code=0000
                    ****    Detailed Statistics Follow    ****
                    Relative Change:
                          From Derivation 1:                1%
                          From Derivation 2:               66%
                          In Common (both Derivations):     1%
                          Net Change:                       1%
                    Inserts:
                          Total Records:                 1614
                          Blocks from Derivation 1:         1  Records per block: Min:      1 Max:      1 Avg:      1
                          Blocks from Derivation 2:       341  Records per block: Min:     48 Max:    202 Avg:      4
                          Blocks in Common (from both):     0  Records per block: Min:      0 Max:      0 Avg:      0
                    Deletes:
                          Total Records:                  945
                          Blocks from Derivation 1:         0  Records per block: Min:      0 Max:      0 Avg:      0
                          Blocks from Derivation 2:       353  Records per block: Min:      1 Max:     44 Avg:      2
                          Blocks in Common (from both):     1  Records per block: Min:      1 Max:      1 Avg:      1
                    Conflicts:
                          Total Conflict Areas:             0
                          Derivation 1 records:             0
                          Derivation 2 records:           944
                          *Contention area detected at WIP record number 002767
                          Total Contention Areas:           1
                    BUILD WIP Summary:
                          Total records in WIP:          3883
                          Complexity Factor:                1
                    ****     End of Detail Statistics     ****
```

6.3.1 Overview

The Build WIP Detail report is created when your build requests have been validated successfully. This report provides you with information about each member set involved in the Build WIP process, identifying the WIP, Root, Derivation 1, and Derivation 2 sources and members. Additional detail information is included, unless you use the STATS OFF (DETAIL) option when first entering the request. If you enter the STATS OFF (ALL) option, this report is not produced at all.

There are three parts to each member set description:

- Beginning of Build WIP information
- Return code information
- Detail statistics

6.3.2 PDM Build WIP Detail Report Fields Description

6.3.2.1 Beginning Build of WIP Information

The first section, BEGINNING BUILD OF WIP, indicates the beginning of the Build WIP process *for this member only*. This line appears as the first line whenever information for a new member is presented. This section provides the following data for each member set:

Field	Description
WIP	The location of the WIP data set.
Member	The WIP member name.
Root	The location of the Root source, either a data set or an Endevor location.
Member	The Root member or element name.
Derivation 1	The location of the Derivation 1 source, either a data set or an Endevor location.
Member	The Derivation 1 member or element name.
Derivation 2	The location of the Derivation 2 source, either a data set or an Endevor location.
Member	The Derivation 2 member or element name.

6.3.2.2 Return Code Information

The second section of the member set description begins with a time stamp and the informational message PDM0064I, which indicates the generation of a **return code** for the member set. The return code listed indicates whether the WIP file was built successfully; a return code of 0000 indicates that the WIP file was built with no problems.

A return code higher than zero indicates that an error has occurred. A second error message should appear also, providing you with additional information. In this situation, the WIP member built will most likely be invalid.

6.3.2.3 Detail Statistics

The third section contains **detail statistics** for the member set. This section appears for every member unless you have selected the STATS OFF (DETAIL) option.

Relative Change: Provides information about the number of changes made by both Derivations, relative to the original (Root) module:

Field	Description
From Derivation 1	Indicates the percent of the original module changed by Derivation 1.
From Derivation 2	Indicates the percent of the original module changed by Derivation 2.
In Common	Indicates the percent of the original module changed by both Derivations at the same (common) place in the program.
Net Change	The total percent by which the original module has changed.

Inserts and Deletes: The Inserts and Deletes sections provide information about the number of insert/delete lines from both Derivations.

Field	Description
Total Records	Indicates the total number of lines marked as inserts/deletes into the original module by both Derivation 1 and Derivation 2.
Blocks from Derivation 1	A *block* is a group of consecutive inserted/deleted lines from one Derivation. For example, if Derivation 1 inserts/deletes 10 lines in one place, that group of 10 lines constitutes a block, with a block size of 10. Similarly, if Derivation 2 inserts/deletes 5 lines in one place, that group of 5 lines constitutes a block, with a block size of 5. BLOCKS FROM DERIVATION 1 indicates the number of blocks of inserted/deleted lines from the Derivation 1 file only. RECORDS PER BLOCK provides additional detail, by noting: ■ MIN — Smallest block size ■ MAX — Largest block size ■ AVG — Mean average block size from the Derivation 1 file only.
Blocks from Derivation 2	Indicates the number of blocks of inserted/deleted lines from the Derivation 2 file only. RECORDS PER BLOCK provides additional detail, by noting: ■ MIN — Smallest block size ■ MAX — Largest block size ■ AVG — Mean average block size from the Derivation 2 file only.
Blocks in Common (from both)	Indicates the number of blocks of inserted/deleted lines, made by both Derivations, in common (that is, inserts/deletes at the same location, with respect to the Root program). RECORDS PER BLOCK provides additional detail, by noting: ■ MIN — Smallest block size ■ MAX — Largest block size ■ AVG — Mean average block size with inserted/deleted lines in common.

Conflicts The Conflicts section provides information about the number of conflict areas and contention areas found in this particular WIP member. Pay particular attention to this section of the report, as the number of conflicts and contentions can help you assess the complexity of a project.**:** A *conflict area* is a place within the WIP File where both Derivation 1 and Derivation 2 have inserted lines at exactly the same place in their respective copies of the Root program. The first three fields in this section refer to conflict areas only.

Field	Description
Total Conflict Areas	Indicates the total number of conflict areas in this WIP member.
Derivation 1 Records	Indicates the number of Derivation 1 **records** (not blocks) that have "participated" in conflicts in this WIP member.
Derivation 2 Records	Indicates the number of Derivation 2 **records** (not blocks) that have "participated" in conflicts in this WIP member.

A *contention area* is a place within the WIP File at which both Derivations have inserted a block of changes, but the changes do not begin at exactly the same place in the copies of the Root program. Since these blocks overlap, they are considered conflicting in nature. You should always review contention areas for potential conflicts. The remaining fields in this section of the report refer to contention areas only.

Field	Description
Contention Area Detected at WIP Record Number	Indicates the beginning line, within the WIP file, of a contention area.
Total Contention Areas	Indicates the total number of contention areas in this WIP member.

Build WIP Summary Summarizes information for this particular WIP member.

Field	Description
Total Records in WIP	Indicates the total number of records contained in this WIP File from: ■ Root ■ Derivation 1 ■ Derivation 2

Field	Description
Complexity Factor **1**	A value assigned to the WIP file. The complexity factor is a function of the number of: ■ Records in (size of) the WIP file. ■ Derivation 1 records in conflict. ■ Derivation 2 records in conflict. ■ Conflict areas and contention areas. The complexity factor ranges from **0-5:** ■ 0 — No changes occurred in the member. ■ 1 — Changes occurred, but there are no conflict or contention areas. ■ 2-5 — The member contains conflicts, where – 2 — Signifies simple conflicts – 5 — Signifies complex conflicts

Note: **1** : Two members may have the same complexity factor, but may be radically different in regard to the nature of the conflicts. Therefore, you should pay careful attention to the complexity factor for each WIP member, and should review those members with factors of 2 or above.

6.4 PDM Build WIP Summary Report

```
COPYRIGHT (C) Computer Associates, INC., 2002                                    24JAN02 09:53:26       PAGE   1
                                Parallel Development Option BUILD WIP Summary Report          RELEASE x.x  SERIAL Xnnnn
 WIP Data Set.... ENDEVOR.PDM.WIPLIB
 Root............ ENDEVOR.NQAPROD2.SRCLIB
 Derivation 1.... ENDEVOR /P/NDVR /PDM     /ASMIPGMR
 Derivation 2.... ENDEVOR.PDM.SRCLIB
 WIP      Root               Complexity Derivation 1     Der 1   Der 1   Derivation 2     Der 2   Der 2   Common  Common  Auto-
 Member   Member           RC Factor     Member           Inserts Deletes Member           Inserts Deletes Inserts Deletes Merge
 -------- ---------------- -- ---------- ---------------- ------- ------- ---------------- ------- ------- ------- ------- -----
 BC1G2500 BC1G2500          0     5      BC1G2500   01.29      44       2 BC1G2500            2718    1306       5      23   N
 BC1G2100 BC1G2100          0     3      BC1G2100   01.16     300       3 BC1G2100             537     176     280     439   N
 BC1G2000 BC1G2000          0     1      BC1G2000   01.31       1       0 BC1G2000            1613     944       0       1   N
 BC1G2200 BC1G2200          0     1      BC1G2200   01.03       0       0 BC1G2200              41      35       0       0   N
 BUILD WIP Summary
   Number of members:
     Selected for processing........    4
     Successfully processed.........    4
     With REPLACE conflicts.........    0
     In error.......................    0
     With no inserts or deletes.....    0
     With no conflicts..............    2
     Automatically Merged...........    0
     With Complexity Factor 5.......    1
     With Complexity Factor 4.......    0
     With Complexity Factor 3.......    1
     With Complexity Factor 2.......    0
     With Complexity Factor 1.......    2
     With Complexity Factor 0.......    0
   Highest BUILD return code........    0
```

6.4.1 Overview

The PDM Build WIP Summary Report summarizes the detail information provided on the PDM Build WIP Detail Report, even if you suppressed detail information from printing. Data for the entire WIP data set is presented.

Warning: If you code the STATS OFF ALL statement, this report is not produced.

6.4.2 PDM Build WIP Summary Report Field Descriptions

6.4.2.1 WIP Data Set Library Information

This section of the report identifies the **WIP data set** and the components of the WIP data set, the Root and Derivation files.

Field	Description
WIP Data Set	Identifies the WIP data set constructed from the BUILD WIP process.
Root	Identifies the base Root file of this WIP data set.
Derivation 1	Identifies the first Derivation data set used during Build processing.
Derivation 2	Identifies the second Derivation data set used during Build processing.

WIP Data Set Member Information This section of the report displays specific information regarding the WIP, Root, and Derivation files used during the BUILD WIP process.

Field	Description
WIP Member	Identifies the WIP data set member.
Root Member	Identifies the Root member of the WIP data set.
RC	Identifies the return code generated from the BUILD WIP process.
Complexity Factor	A value assigned to the WIP file. The complexity factor is a function of the number of: ■ Records in (size of) the WIP file. ■ Derivation 1 records in conflict. ■ Derivation 2 records in conflict. ■ Conflict areas and contention areas. The complexity factor ranges from **0-5:** ■ 0 — No changes occurred in the member. ■ 1 — Changes occurred but there are no conflict or contention areas. ■ 2-5 — Indicates that the member contains conflicts, where: – 2 — Signifies simple conflicts – 5 — Signifies complex conflicts
Derivation 1 Member	Identifies the Derivation 1 member (or Endevor element, including version and level numbering) of the WIP data set.
Der 1 Inserts	Displays the number of records Derivation 1 added to the Root member.
Der 1 Deletes	Displays the number of records Derivation 1 deleted from the Root member.
Derivation 2 Member	Identifies the Derivation 2 member (or Endevor element, including version and level numbering) of the WIP data set.
Der 2 Inserts	Displays the number of records Derivation 2 added to the Root member.
Der 2 Deletes	Displays the number of records Derivation 2 deleted from the Root member.
Common Inserts	Displays the number of common inserts made by both Derivation 1 and Derivation 2 to the Root member.
Common Deletes	Displays the number of common deletes made by both Derivation 1 and Derivation 2 to the Root member.
Auto-Merge	States whether PDM invoked the Automerge feature during the BUILD WIP process.

BUILD WIP Summary Information This section of the report provides an accounting of specific information for all members in the WIP data set.

Field	Description
Selected for Processing	Indicates the total number of members selected for processing.
Successfully Processed	Indicates the number of members processed successfully; that is, with return codes of zero.
With Replace Conflicts	Indicates the number of situations in which there were like-named members in the WIP data set and the replace/no replace option was set to "no replace." The following message that appears to the right of this field: `(PROCESSING BYPASSED)`
In Error	Indicates the number of members that could not be processed due to a problem, such as an I/O or system error. The following message that appears to the right of this field: `(PROCESSING INCOMPLETE)` When a WIP member is created but flagged as "in error," that member should be considered invalid.
With No Inserts or Deletions	Indicates the number of members that had neither insertions nor deletions from either Derivation. Such members have a complexity factor of zero.
With No Conflicts	Indicates the number of members containing changes, but did not have any conflicts resulting from the changes. Such members have a complexity factor of 1 assigned. If these changes involve a contention area, a complexity factor of 2 is assigned.
With Complexity Factor 5-0	Groups the members of the WIP data set by complexity factor, in descending order. The total number of members with a complexity factor of 5 is listed first, followed by the total number of members with a complexity factor of 4, the total with a complexity factor of 3, and so on. As mentioned above, you should pay particular attention to those members with a complexity factor of 2 or above, as these factors indicate that there definitely are conflicts within specific members of the WIP data set. A factor of 5 indicates either an especially complex set of conflicts or a large amount of conflicts involved. The lower the factor, the less complex and fewer number of conflicts. Keep in mind, however, that the same complexity factor can signify different types of conflict situations. A complexity factor of 1 indicates that the member had changes, but there were no conflicts. You should still review this member, however. A complexity factor of 0 indicates that there no changes at all to the member.

6.4.3 Highest BUILD Return Code

The last line of the report indicates the highest BUILD return code. The return code listed here reflects the highest return code encountered during processing of the entire WIP data set.

If you receive a return code other than zero, check the PDM Build WIP Summary Report to find out which member had that return code and where the member is detailed in the PDM Build WIP Detail Report.

If the Create WIP Never clause was specified on the Build action, the following message appears after the highest BUILD return code:

Note: The WIP file(s) were not created at the user's request.

6.5 PDM Syntax Request Report for Merge

```
COPYRIGHT (C) Computer Associates, INC., 2002                                    24JAN02 09:57:17       PAGE   1
                                  Parallel Development Option Syntax Request Report           RELEASE x.x  SERIAL Xnnnn
         09:57:17  PDM0200I  PDM control statement parsing is beginning
          MERGE OUTPUT DSNAME 'ENDEVOR.PDM.MERGELIB'
                WIP    DSNAME 'ENDEVOR.PDM.WIPLIB'
                   REPLACE
                     .
            WIP BC1G2*
            OUT BC1G2*
                     .
         09:57:17  PDM0201I  PDM control statement parsing has successfully completed
```

6.5.1 Overview

When you enter your Merge requests, the Parallel Development Option validates the syntax in the Merge request data set. This PDM Syntax Request Report mirrors exactly what you entered, in the order in which it was entered, and flags any syntax errors.

When all Merge syntax requests have been checked, PDM looks for errors. If no errors can be found, processing continues. If errors do exist, processing is terminated at this point. Refer to the *Error Codes and Messages Guide* for an explanation of any error messages received.

Note the last line in the PDM Syntax Request Report above; this message indicates that all request cards have been processed successfully.

6.6 PDM Merge Output Detail Report

```
COPYRIGHT (C) Computer Associates, INC., 2002                                    24JAN02 09:57:19       PAGE   1
                          Parallel Development Option MERGE Output Detail Report          RELEASE x.x  SERIAL Xnnnn
                        *----------------------------------------*
                        *    Beginning MERGE of WIP member       *
                        *----------------------------------------*
                        WIP:            ENDEVOR.PDM.WIPLIB
                        Member:         BC1G2000
                        OUTPUT:         ENDEVOR.PDM.MERGELIB
                        Member:         BC1G2000
        09:57:21  PDM0064I  MERGE operation is complete.  Return Code=0000
                        ****    Detailed Statistics Follow    ****
                              Total records in WIP:           3867
                              Total records in output:        2922
                              Total invalid WIP records:         0
                        ****     End of Detail Statistics     ****
                        *----------------------------------------*
                        *     Beginning MERGE of WIP member      *
                        *----------------------------------------*
                        WIP:            ENDEVOR.PDM.WIPLIB
                        Member:         BC1G2100
                        OUTPUT:         ENDEVOR.PDM.MERGELIB
                        Member:         BC1G2100
        09:57:23  PDM0064I  MERGE operation is complete.  Return Code=0000
                        ****    Detailed Statistics Follow    ****
                              Total records in WIP:           2947
                              Total records in output:        2329
                              Total invalid WIP records:         0
                        ****     End of Detail Statistics     ****
                        *----------------------------------------*
                        *     Beginning MERGE of WIP member      *
                        *----------------------------------------*
                        WIP:            ENDEVOR.PDM.WIPLIB
                        Member:         BC1G2200
                        OUTPUT:         ENDEVOR.PDM.MERGELIB
                        Member:         BC1G2200
        09:57:25  PDM0064I  MERGE operation is complete.  Return Code=0000
        09:57:30  PDM0080I      3 statistic records were written to dataset ENDEVOR.PDM.STATS
```

6.6.1 Overview

The Merge Output Detail report is created when the Merge request cards have been parsed successfully. This report, like the PDM Build WIP Detail report, provides you with location (source and member) information about each member set involved in the Merge process. Additional detail information is included, unless you code the STATS OFF (DETAIL) option when first entering the requests. Or, if you enter the STATS OFF (ALL) option, this report is not produced at all.

There are three parts to each member set description.

- Beginning merge of WIP member information
- Return code information
- Detail statistics

6.6.2 PDM Merge Output Detail Report Field Descriptions

6.6.2.1 Beginning Merge of WIP Member Information

The first section, BEGINNING MERGE OF WIP MEMBER, indicates the beginning of the Merge process *for this member only*. This line appears as the first line whenever information for a new member is presented. This section provides the following data for each member set:

Field	Description
WIP	The location of the WIP data set.
Member	The WIP member name.
Output	The location of the output source file.
Member	The output source member name.

6.6.2.2 Return Code Information

The second section of the member set description indicates the **return code** for the member set. The return code listed indicates whether the WIP member was merged successfully; a return code of 0000 indicates that the merge was indeed successful.

A return code higher than zero indicates that an error has occurred during processing. A second error message should also appear, providing you with additional information. In this situation, the merged WIP member will most likely be invalid.

6.6.2.3 Detail Statistics

The third section contains **detail statistics** for the member set. This section appears for every member unless you have selected the STATS OFF (DETAIL) option for your Merge requests. This section provides the following data:

Field	Description
Total Records in WIP	Indicates the total number of records contained in this WIP File.
Total Records in Output	Indicates the total number of records in the Merge Output file.
Total Invalid WIP Records	Indicates the number of invalid WIP records detected in this particular WIP member. Invalid records occur when invalid editing changes are made during the WIP editing process. For example, a character other than an asterisk or percent sign mistakenly may have been coded in the first column of the WIP File annotation section.

6.7 PDM Merge Output Summary Report

```
COPYRIGHT (C) Computer Associates, INC., 2002                                    24JAN02 09:57:29       PAGE   1
                    Parallel Development Option MERGE Output Summary Report         RELEASE x.x  SERIAL Xnnnn
                              WIP Data set........... ENDEVOR.PDM.WIPLIB
                              MERGE Output Data set.. ENDEVOR.PDM.MERGELIB
                                 WIP        MERGE          Number of Records  Invalid  WIP Member
                                Member      Member    RC    WIP      MERGE   Records  Disposition
                              ----------  ----------  --  -------   -------  -------  ----------
                              BC1G2000    BC1G2000     0     3867      2922        0  KEPT
                              BC1G2100    BC1G2100     0     2947      2329        0  KEPT
                              BC1G2200    BC1G2200     0      289       254        0  KEPT
                              BC1G2500    BC1G2500     0     6063      4732        0  KEPT
                              MERGE Summary:
                               Number of members:
                                 Selected for processing........     4
                                 Successfully processed.........     4
                                 Deleted from the WIP dataset...     0
                                 With REPLACE conflicts.........     0  (Not processed)
                                 In error.......................     0
                                Highest MERGE return code .....    00
```

6.7.1 Overview

The PDM Merge Output Summary report summarizes the detail information provided on the PDM Merge Output Detail report, even if you suppressed detail information from printing. Data for all WIP and output source files is presented.

Warning: If you code the STATS OFF ALL statement, this report is not produced.

6.7.2 PDM Merge Output Summary Report Field Descriptions

6.7.2.1 WIP Data Set Library Information

The first line of this report identifies the **WIP data set** for which the Merge process was run.

6.7.2.2 Output Data Set Information

The second line of this report identifies the **MERGE output data set** in which the merged output is stored.

6.7.2.3 WIP Member Information

This section of the report provides information about each WIP member that was merged into the MERGE output data set.

Field	Description
WIP Member	Identifies the WIP member for which the Merge process was run.
Merge Member	Identifies the Merge output member created by the Merge process.

Field	Description
RC	Displays the return code for the Merge process for each WIP/Merge member.
Number of Records	Displays the number of records in the WIP data set member and the Merge output data set member.
Invalid Records	Displays the number of records from the WIP data set that could not be processed.
WIP Member Disposition	Indicates whether the WIP member has been **KEPT** or **DELETED** after completing the MERGE process.

6.7.2.4 Merge Summary Information

The next section of the report provides summary information about all the members in the WIP data set:

Field	Description
Selected for Processing	Indicates the total number of members selected for processing.
Successfully Processed	Indicates the number of members processed successfully; that is, with return codes of zero.
Deleted from the WIP data set	Indicates the difference between the WIP members that were selected for processing and those that were actually processed successfully. Members that could not be processed are deleted from the WIP data set.
With Replace Conflicts	Indicates the number of situations in which there were like-named members in the WIP data set and the replace/no replace option was set to "no replace." Such members are not processed. The following message appears to the right of this field: `(Not Processed)`
In Error	Indicates the number of members that could not be processed due to a problem, such as an I/O or system error. The following message appears to the right of this field: `(Processing Incomplete)` Members flagged as "in error" should be considered invalid.

6.7.3 Highest MERGE Return Code

The last line of the report indicates the highest merge return code. The return code here reflects the highest return code that occurred during the merge of the entire WIP data set.

Appendix A. Batch Execution JCL

A.1 Sample Batch Execution JCL

When you use batch processing for the Build WIP and/or Merge functions, you must submit the resulting batch requests for execution. You can submit these requests using either the submit option from the or by creating your own JCL.

A.1.1 Before Executing JCL

If you use your own JCL, be sure to do the following prior to execution:

- Construct a valid JOBCARD.
- Ensure that the CONLIB data set name is correct.
- Specify the batch request cards to be used.

A.1.2 Sample

Sample batch execution JCL follows. The JCL can be found in **iprfx.iqual.JCLLIB** (BC1GJCL2).

```
//*  ( COPY JOBCARD )
//*********************************************************************
//*                                                                   *
//*    BC1GJCL2 - PDM BATCH EXECUTION JCL                             *
//*                                                                   *
//*    THE FOLLOWING CHANGES MUST BE MADE TO THIS JCL BEFORE IT CAN   *
//*    BE EXECUTED:                                                    *
//*                                                                   *
//*    1. ADD THE APPROPRIATE JOB CARD                                *
//*    2. IF YOU ARE A STANDALONE PDM USER, THAT IS, YOU ARE NOT      *
//*       RUNNING WITH ENDEVOR, THEN DELETE THE EXEC STATEMENT THAT*
//*       EXECUTES PROGRAM NDVRC1 AND UNCOMMENT THE STATEMENT THAT    *
//*       EXECUTES PROGRAM BC1G0000.                                  *
//*    3. REVIEW THE CONLIB DD AND THE STEPLIB DD STATEMENT AND REFER *
//*       TO THE CORRECT PDM.CONLIB DATASET.                          *
//*    4. MODIFY THE BATCHIN DD STATEMENT TO POINT TO THE FILE THAT   *
//*       CONTAINS YOUR PDM REQUEST CONTROL CARDS.
//*    5. IF THE PDM SUMMARY REPORTS ARE TO BE WRITTEN TO A           *
//*       LOCATION THAN THE PDM DETAIL REPORTS THEN                   *
//*       UNCOMMENT THE C1MSGS2 DD STATEMENT AND MAKE THE             *
//*       APPROPRIATE CHANGES                                         *
//*                                                                   *
//*********************************************************************
//PDM      EXEC PGM=NDVRC1,PARM='BC1G0000',DYNAMNBR=1500,REGION=3096K
//*PDM     EXEC PGM=BC1G0000,DYNAMNBR=1500,REGION=3096K
//STEPLIB  DD DSN=iprfx.iqual.PDM.CONLIB,DISP=SHR
//*        DD DSN=SYS2.PANVALET.LOAD,DISP=SHR       PANVALET LOADLIB
//CONLIB   DD DSN=iprfx.iqual.PDM.CONLIB,DISP=SHR
//C1MSGS1  DD SYSOUT=*
//*C1MSGS2  DD SYSOUT=*                             SUMMARY REPORT
//SYSABEND DD SYSOUT=*
//BATCHIN  DD DSN=iprfx.iqual.PDM.SYNTAX,DISP=SHR
```

Appendix B. Batch Syntax

B.1 PDM Syntax

PDM syntax is a free-form language consisting of English-like statements. The syntax allows you to drive a PDM Build WIP or Merge batch process quickly and easily.

You can generate PDM syntax request statements by using the PDM Build and Merge dialogs. You can also code the syntax requests yourself. The goal of PDM syntax is to provide you with a flexible yet easy-to-use method for creating your requests and, subsequently, submitting them for processing.

This appendix illustrates and describes how to use PDM syntax.

B.2 PDM Syntax Statements

PDM syntax involves only two actions — Build and Merge. PDM syntax is entered in, and referred to as, *statements*. A PDM syntax statement ends with a period (.).

A statement consists of one or more *clauses*. A clause is an individual line of information within each statement (for example, root DSN "root.lib" or compare 007 through 072). One statement may contain several clauses.

There are three types of statements you can use when coding each action:

- **The ACTION Statement** — The action statement defines the action you want to perform against a set of elements or members, as well as the location of those elements or members. Action statements are the only statements actually executed by PDM, and only one action statement (of either type) is required.

  The action statements are *BUILD* and *MERGE*.

- **The MASK Statement** — The mask statement defines the member(s) or element(s) against which the action will take place. You can enter either an explicit member or element name, or you can use a name mask. You can enter as many mask statements as necessary for each action statement.

- **The STATS OFF Statement** — Statistical reports are generated for both the Build WIP and Merge processes. You have the option of suppressing these reports entirely, or limiting the amount of detail printed. The STATS OFF statement allows you to determine how detailed a report you want to print.

B.2.1 The Structure of a PDM Request Statement

When building a PDM request statement:

- **You must always include an ACTION statement.** If you enter a BUILD statement, you must also enter WIP, Root, and Derivation 1 location information. Optionally, you can add Derivation 2, compare columns, and replace/no replace data.

- **You must always include BUILD MASK or MERGE MASK statements** (unless you are using sequential files). MASK statements, defining the members and/or elements designated in the ACTION statement, must follow that ACTION statement (unless you are dealing with sequential files). A MASK statement entered with no preceding ACTION statement is flagged as an error.

- **STATS OFF statements are optional.** If you enter a STATS OFF statement, it must follow the ACTION statement to which it pertains. If you enter a STATS OFF statement before the related ACTION statement, you receive an error.

Note: The positioning of the MASK statement and STATS OFF statement is interchangeable. It does not matter which statement immediately follows the ACTION statement, as long as the ACTION statement is coded first.

B.2.2 Process Flow

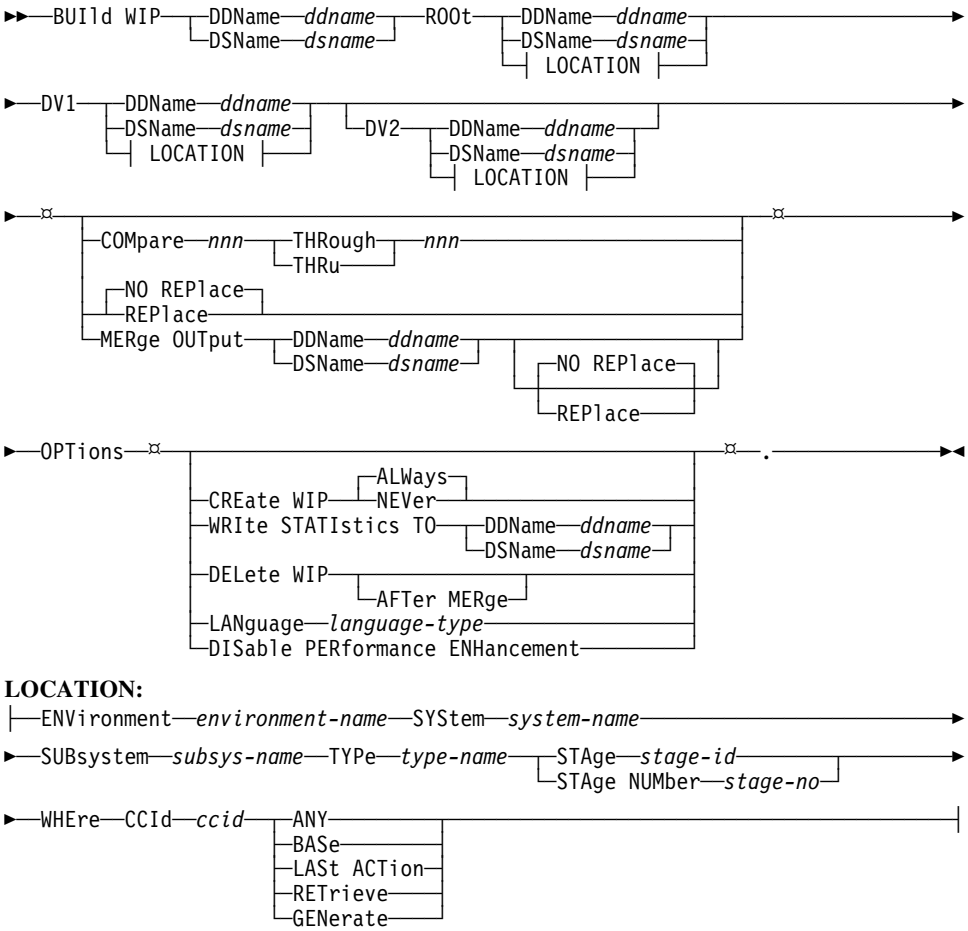
When you submit your Build WIP or Merge requests, PDM follows a specific processing flow:

1. PDM first parses, or validates, the syntax.
2. A Build WIP or Merge Syntax Report is produced, echoing the Build WIP or Merge statements entered, and flagging any syntax errors.
3. When all requests have been validated, PDM checks for errors. If any errors exist within the syntax, processing terminates.

   If no errors exist, PDM continues processing by calling in the appropriate Build WIP or Merge routine.

B.3 The BUIld WIP Statement

BUIld WIP Syntax:



LOCATION:

B.3.1 Parameters

The rules listed below provide information pertaining to each clause of the Build action statement.

BUIld

The keyword BUIld must be the first word in a Build action statement and it is required.

WIP

You specify the WIP data set by providing one of these values:

- DDName - requires the appropriate JCL
- DSName (data set name)

This parameter is required.

ROOt

You specify the Root location by providing one of the following:

- DDName
- DSName
- Endevor location - requiring either:
  - ENVironment
  - SYStem
  - SUBsystem
  - TYPE
  - And either:
    - STAge
    - STAge NUMber

  OR
  - Provide a CCID using WHEre CCId and optionally specify:
    - ANY (default search criterion)
    - BASE
    - LASt ACTion
    - RETrieve
    - GENerate

This parameter is required.

DV1

Specifies Derivation 1 location by providing one of these values:

- DDName - requires the appropriate JCL
- DSName (data set name)
- LOCATION

This is a required parameter.

DV2

Specifies Derivation 2 location by providing one of these values:

- DDName - requires the appropriate JCL
- DSName (data set name)
- LOCATION

This is an optional parameter.

Note: The maximum LRECL allowed for a Derivation 1 or Derivation 2 member is 247.

COMpare

Identifies the columns for comparison during the Build WIP process process, to establish changes. This is an optional parameter. If selected, you must provide a beginning column and an ending column. If only one column value is entered, you will receive an error message.

(compare from) nnn

Indicates the beginning column for the comparison. Default is 7.

(through value) nnn
Indicates the ending column for the comparison. Default is 72.

Note: Use caution when specifying a compare range for variable length records. If record lengths are different, PDM may annotate a record (line) as changed, even if record data is identical. For example, trailing blanks may be truncated in a variable length record.

REPlace/NO REPlace
Indicates on a global level, whether you want to replace like-named WIP members in the WIP data set. This is an optional parameter.

REPlace
Replaces existing members with like-named new members.

NO REPlace
Specifies PDM will not replace like-named WIP members in the WIP data set, PDM will not write the new member to the WIP data set. NO REPlace is the default.

MERge OUTput
Identifies the location for the merge output library associated with this WIP data set. This parameter is optional. If you code this clause, you must provide:

- DDName - requires the appropriate JCL
- DSName (data set name)

REPlace
Replaces existing members with like-named new members.

NO REPlace
Specifies PDM will not replace like-named WIP members in the WIP data set, PDM will not write the new member to the WIP data set. NO REPlace is the default.

OPTions
Use this option to indicate whether PDM is to create a WIP file during the Build process. Acceptable values are:

ALWAYS
Default. Indicates that PDM is to create a WIP file during the Build process.

NEVER
Indicates that PDM is not to create a WIP file.

WRIte STATIstics TO
This parameter denotes the location of the statistics file gathered during the Build process. Valid values are:

- DDName - requires the appropriate JCL
- DSName (data set name)

This parameter is optional. Refer to Appendix D, “Statistical Data Control Blocks” for information on statistics data set characteristics.

DELete WIP
This parameter specifies the WIP file is deleted after successful completion of the Merge process.

LANguage
Specifies the associated language type for the members when PDM automatically merges the WIP file and the merge data set is an AllFusion CA-Panvalet or AllFusion CA-Librarian data set.

AllFusion CA-Panvalet
Valid values are:

- ALC
- ANSCOBOL
- AUTOCODE
- BAL
- COBOL
- COBOL-72
- DATA
- FORTRAN
- JCL
- OBJECT
- PL/1
- RPG
- USER180
- USER780
- OTHER

AllFusion CA-Librarian
Valid values are:

- ASM
- COB
- DAT
- FOR
- FRG
- FRS
- GIF
- GOF
- JCL
- PLF
- PLI
- RPG
- TXT
- VSB

DISable PERformance ENHancement
Directs PDM to use external (disk) storage instead of virtual storage for Build WIP processing. Generally, this option is not specified unless necessitated by message PDM2001E. Specifying this option degrades Build WIP processing performance.

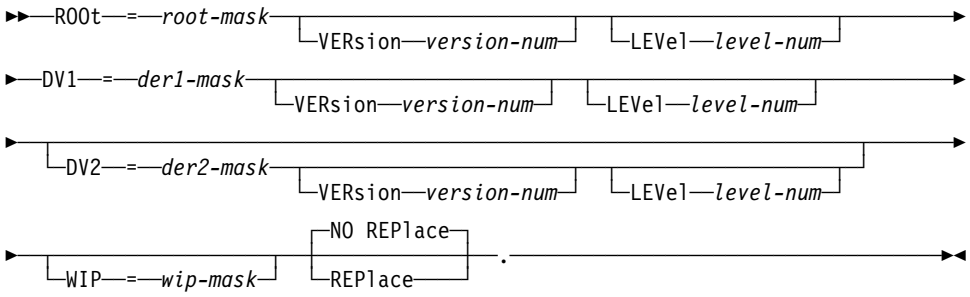
Notes:

1. For sequential file, no additional information is required for the BUILD statement, unless you want to use the STATS OFF option, which is described later.
2. For partitioned data sets, AllFusion CA-Panvalet, AllFusion CA-Librarian files, or Endevor locations, you must enter the appropriate MASK statements.

B.4 The BUIld MASK Statement

The BUILD MASK statement defines each set of members or elements for which a WIP File is to be built. You can enter as many BUILD MASK statements as are required. BUILD MASK statements must follow the BUILD statement.

BUIld Mask Syntax:



B.4.1 Parameters

ROOt

Allows you to designate one or more members or elements by providing one of the following:

- An explicit member or element name
- A name mask
- An Endevor location and optionally, a specific version and level for that element

This parameter is required.

DV1

Allows you to designate one or more members or elements by providing one of the following:

- An explicit member or element name
- A name mask

This parameter is required.

DV2

Allows you to designate one or more members or elements by providing one of the following:

- An explicit member or element name
- A name mask

This parameter is optional.

WIP

Defines the WIP member to be built. This parameter is optional.

wip-mask

This can be:

- An explicit name.
- Blank. If you do not specify a *wip-mask* in this MASK statement, the WIP member name defaults to the Root member/element name.
- An overlay mask. An overlay mask differs from the standard name mask in that the characters specified (before the asterisk) overlay the corresponding characters in the Root member name.

For example, you enter a Root name of BC1PAL10 and a WIP mask of W*. The WIP name for this member is WC1PAL10.

REPlace/NO REPlace

Allows you to override the global replace/no replace setting in the BUILD statement for the member set defined in this MASK statement only. This parameter is optional and if it is not specified, the global setting from the BUILD statement takes precedence.

REPlace

Replaces existing members with like-named new members.

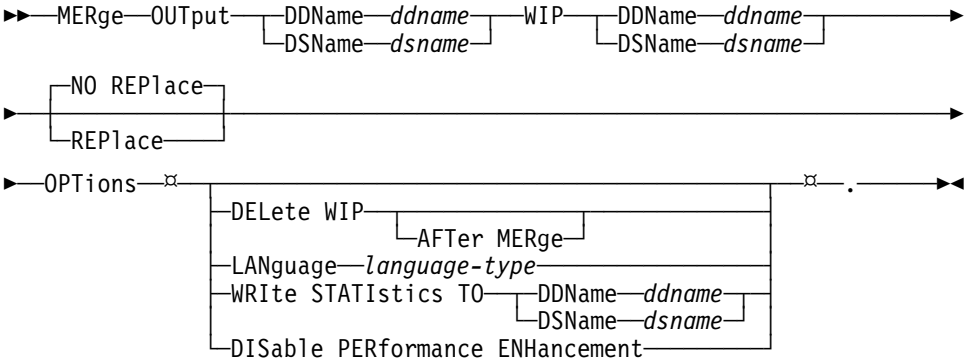
NO REPlace

Specifies PDM will not replace like-named WIP members in the WIP data set, PDM will not write the new member to the WIP data set. NO REPlace is the default.

B.5 The MERGE Statement

Use the MERGE statement to enter requests to merge the WIP File into an output source file.

MERge OUTput Syntax:



B.5.1 Parameters

MERge
The keyword MERge must be the first word in a MERGE statement.

OUTput
Identifies the merge output file. Valid Values are:

- DDName - requires the appropriate JCL
- DSName (data set name)

This is a required parameter.

WIP
You specify the WIP data set by providing one of these values:

- DDName - requires the appropriate JCL
- DSName (data set name)

This parameter is required.

REPlace/NO REPlace
Indicates on a global level, whether you want to replace like-named in the merge output data set. The default is NO REPlace.

REPlace
Replaces existing members with like-named new members.

NO REPlace
Specifies PDM will not replace like-named WIP members in the merge output data set, PDM will not write the new member to the merge output data set. This is the default.

OPTions

Specifies the language type for members, data set for statistics information, and if the WIP file should be deleted.

DELete WIP AFTer MERge

Specifies the WIP file is deleted after successful completion of the merge process.

LANguage

Specifies the associated language type for the members when PDM automatically merges the WIP file and the merge data set is an AllFusion CA-Panvalet or AllFusion CA-Librarian data set.

AllFusion CA-Panvalet

Valid values are:

- ALC
- ANSCOBOL
- AUTOCODE
- BAL
- COBOL
- COBOL-72
- DATA
- FORTRAN
- JCL
- OBJECT
- PL/1
- RPG
- USER180
- USER780
- OTHER

AllFusion CA-Librarian

Valid values are:

- ASM
- COB
- DAT
- FOR
- FRG
- FRS
- GIF
- GOF
- JCL
- PLF
- PLI
- RPG
- TXT
- VSB

If PDM is to automatically merge the WIP file and the merge data set is an AllFusion CA-Panvalet or an AllFusion CA-Librarian data set, you can

specify the language to be associated with the member. The following is a list of language types:

AllFusion CA-Panvalet: ALC, ANSCOBOL, AUTOCODE, BAL, COBOL, COBOL-72, DATA, FORTRAN, JCL, OBJECT, PL/1, RPG, USER180, USER780, OTHER.

AllFusion CA-Librarian: ASM, COB, DAT, FOR, FRG, FRS, GIF, GOF, JCL, PLF, PLI, RPG, TXT, VSB.

Notes:

1. For sequential file, no additional information is required for the MERGE statement, unless you want to use the STATS OFF option, which is described later.
2. For partitioned data sets, AllFusion CA-Panvalet, AllFusion CA-Librarian files, or Endevor locations, you must enter the appropriate MASK statements.

WRIte STATIstices TO

This parameter denotes the location of the statistics file gathered during the MERGE process. Valid values are:

- DDName - requires the appropriate JCL
- DSName (data set name)

This parameter is optional. Refer to Appendix D, "Statistical Data Control Blocks" for information on statistics data set characteristics.

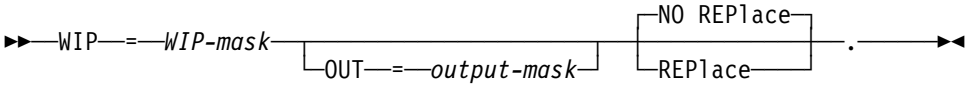
DISable PERformance ENHancement

Use this option to tell PDM to use external (disk) storage instead of virtual storage for Merge processing. Generally, this option is not specified unless necessitated by message PDM2001E. Specifying this option degrades Merge processing performance.

B.6 The MERGE MASK Statement

Use the MERGE MASK statement to define the WIP and output source files involved in the Merge process. You can enter as many MERGE MASK statements as are required. MERGE MASK statements must follow the MERGE action statement.

Merge Mask Syntax:



B.6.1 Parameters

WIP
Defines the WIP member to be merged.

wip-mask
This can be:

- An explicit member name.
- A name mask.

OUT
Identifies the output file to which the WIP File is merged.

out-mask
Can be one of the following entries:

- An explicit member name.
- Blank.
- An overlay mask.

REPlace/NO REPlace
Overrides the global REPlace/NO REPlace switch in the merge statement for the member set defined in this MASK statement only. This parameter is optional. If it is not specified, the global setting from the MERGE statement takes precedence.

REPlace
Replaces existing members with like-named new members.

NO REPlace
Specifies PDM will not replace like-named WIP members in the merge output data set, PDM will not write the new member to the merge output data set. NO REPlace is the default.

B.7 The STATS OFF Statement

The STATS OFF statement controls the generation of the PDM Build and Merge reports. It is an optional statement; but when specified, the STATS OFF statement must follow the related BUILD or MERGE statement. If you enter a STATS OFF statement without a preceding BUILD or MERGE statement, you receive an error message.

Note: The order of the MASK statement and the STATS OFF statement is interchangeable. It does not matter which statement follows the related action statement, as long as the action statement is coded first.

STAts OFF Syntax:

B.7.1 Parameters

STAts OFF
The keywords STAT and OFF must be the first words in a STATS OFF statement and they are required.

Detail Clause
Indicates the level of information you want to suppress:

DETail
Suppresses the detail information provided for each WIP member in the PDM Build WIP Detail Report and/or the PDM Merge Detail Report (depending on the action for which you code this statement). This is the default.

All
Suppresses detail and summary information; that is, no reports are produced. If you select this option, you only receive the final return code in the JCL.

B.8 Syntax Examples

The following examples illustrate the use of PDM syntax to enter Build WIP and Merge requests.

EXAMPLE 1: Enter Build WIP and Merge information using an explicit member name.

```
build wip     Dsn 'endevor.pdm.wiplib'
      root    Dsn 'endevor.pdm.rootlib'
      dv1     Dsn 'endevor.pdm.dv1lib'
      dv2     Dsn 'endevor.pdm.dv2lib'
      comPARE 007 thru 072
      no replace      .
      root    progx
      dv1     progx
      dv2     progx
      wip     progx           .
 merge wip     dsn 'endevor.pdm.wiplib'
      output  dsn 'endevor.pdm.mergelib'
      no replace       .
      wip progx
      out progx .
```

Note the NO REPlace option is globally set for the BUILD and MERGE statements, and is not overridden by either of the MASK statements.

EXAMPLE 2: Enter Build WIP and Merge information using a combination of data set and Endevor location information. In this example, the WIP and Derivation 1 locations are data sets; the Root and Derivation 2 locations are from Endevor.

```
build wip    dsn   'endevor.pdm.wiplib'
      root   env   support
             sys   finance
             sub   acctrec
             type  copybook
             stage b
      dv1    dsn   'endevor.pdm.dv1lib'
      dv2    env   support
             sys   finance
             sub   acctrec
             type  copybook
             stage p
      compare 007 thru 072
      no replace      .
      root   copy1 ver 01 lev 02
      dv1    progx
      dv2    copy1
      wip    progx      .
merge wip    dsn 'endevor.pdm.wiplib'
      output dsn 'endevor.pdm.mergelib'
      no replace             .
 wip progx
      out progx  .
```

The element COPY 1 is assigned as the explicit element name for both the Root and Derivation 2. Note a version and level have been assigned to the Root element name. This indicates the current version/level of the element is used for Derivation 2, and a previous version/level of the element is used for the Root.

The NO REPlace option is assigned globally to the BUILD statement, with no overriding option in the BUILD MASK statement. The MERGE and MERGE MASK statements are the same as in example 1.

EXAMPLE 3: This SCL builds and automatically merges three files. The actions create statistics records and delete the WIP member if the Merge is successful. Remember, the automatic merge is performed only if the WIP file contains no conflicts.

```
BUILD WIP  DSNAME     'ENDEVOR.PDM.WIPLIB '
      ROOT DSNAME     'ENDEVOR.PDM.ROOT '
      DV1  DSNAME     'ENDEVOR.PDM.DV1 '
      DV2  DSNAME     'ENDEVOR.PDM.DV2 '
           COMPARE     001 THRU 072
           REPLACE
      MERGE OUTPUT
              DSNAME 'ENDEVOR.PDM.MERGELIB '
             REPLACE
      OPTIONS DELETE WIP
             WRITE STATISTICS TO DSNAME
              'ENDEVOR.PDM.STATS '
                      .
    WIP  'C1BMEI00  '
    ROOT 'C1BMEI00  '
    DV1  'C1BMEI00  '
    DV2  'C1BMEI00  '
                                    .
    WIP  'C1BMEI10  '
    ROOT 'C1BMEI10  '
    DV1  'C1BMEI10  '
    DV2  'C1BMEI10  '
                                    .
```

Appendix C. Extending the WIP Edit Macro Facility

C.1 Before You Begin

The PDM Edit WIP function, option 2 of AllFusion Endevor Change Manager Parallel Development Option menu, uses the ISPF/PDF Edit Macro facility to implement the WIPCOUNT, WIPLDEL, WIPUNDEL, WIPSHOW, WIPCHANG, WIPCON and WIPPARA commands.

The enhanced WIP edit commands are written in the TSO/E CLIST language and can be easily modified or extended.

The information in this chapter assumes familiarity with the TSO/E CLIST or REXX language, ISPF/PDF concepts and the ISPF/PDF Edit Macro facility. Refer to the appropriate IBM documentation for further information on these facilities.

C.2 Existing Edit Macros

During PDM installation, a set of CLISTs is copied from the PDM installation tape into the site CLIST library. CLIST member BC1GM100 is the PDM Initial Macro and is used to establish the alias mapping between each CLIST name and the corresponding Edit Macro name.

The following table identifies the CLIST member name and the corresponding WIP Edit Macro name:

Member Name	Edit Macro Name
BC1GM101	WIPLDEL
BC1GM102	WIPSHOW
BC1GM103	WIPCON
BC1GM104	WIPCOUNT
BC1GM105	WIPCHANG
BC1GM106	WIPPARA
BC1GM107	WIPUNDEL
BC1GM108	WIPHELP

C.3 Extending the Edit Macros

Because PDM Edit Macros can be written in the TSO/E CLIST or REXX language, it is easy to add additional customized Edit commands. The existing PDM Edit Macros can be used as guides for developing customized Edit Macros.

CAUTION:
Do not modify the Computer Associates-supplied edit macros. If you want to extend the functionality of the supplied edit macros, copy the Computer Associates edit macro to a new macro, and make the desired changes to the new macro.

C.3.1 BC1GM100 and BC1GMU01

The PDM Edit WIP Initial Edit Macro, BC1GM100, executes a second CLIST, BC1GMU01, during the edit session startup. BC1GMU01 is available to the installation to perform additional, customized startup functions.

BC1GMU01 can be used, for example, to establish aliases for site customized edit macros or to issue ISPF/PDF Edit commands. The version of BC1GMU01 supplied is a skeleton that does not issue any commands and returns a return code of zero. BC1GMU01 can be enhanced using either the CLIST or REXX language.

Appendix D. Statistical Data Control Blocks

D.1 General Information

Read this appendix for information about generating and interpreting PDM information.

PDM can optionally create data records containing detail and summary information about Build or Merge actions. The statistical records can subsequently be processed by a user program.

To have PDM create these statistics, you must code the WRITE STATISTICS TO clause with the Build or Merge action.

PDM can create statistical information about the Build WIP and Merge operations. These statistics can then be used to generate customized reports.

PDM writes the statistics to the location specified in the WRITE STATISTICS TO clause of the BUILD and MERGE statements. The statistics data set must have the following attributes:

```
DCB=(LRECL=540,RECFM=VB,BLKSIZE=block_size,DSORG=PS)
```

The block size can be any value greater than or equal to the LRECL value plus four.

Note: PDM allocates the data set specified in the WRITE STATISTICS TO DSNAME clause with a disposition of OLD. Therefore, any existing data in the statistics data set is overwritten.

If you are performing multiple build or merge operations in a single job, PDM validates the data set to ensure that it has the correct attributes. PDM writes the statistical data in one of the following formats:

- **$PBWSTAT** - WIP Detail Statistics
- **$PBWSSUM** - WIP Summary Statistics
- **$PBMSTAT** - Merge Detail Statistics
- **$PBMSSUM** - Merge Summary Statistics

These control blocks correspond to the PDM Build WIP and Merge Output Detail and Summary Reports.

This appendix provides the record layouts for the four control blocks and explains all data contained within each block.

D.2 $PBWSTAT: WIP Detail Statistics

The $PBWSTAT control block presents detail information about each member of the WIP data set, and reflects much of the statistical information provided on the PDM Build WIP Detail Report. One $PBWSTAT control block is written for each WIP member set.

```
*--------------------------------------------------------------------*
* MACRO: $PBWSTAT                                                    *
*                                                                    *
* FUNCTION: The $PBWSTAT MACRO maps the PDM BUILD WIP Statistics     *
*  record.  One record is created for each WIP file created by the   *
*  BUILD WIP action.                                                 *
*--------------------------------------------------------------------*
$PBWSTAT DSECT                         BUILD WIP Statistics record
$PBWLEN  DC    Y($PBWDSLN)             Length of the Statistics record
$pbwece  dc    cl4'stat'               eyecatcher
$PBWVERS DC    Y(1)                    Version Number
PWTYPE   DC    Y(1)                    Record Type: WIP Statistics
PWREPL   DS    CL1                     Global replace option (Y/N)
PWPRCMDE DS    CL1                     Processing Mode (F/B)
PWCMPBEG DS    H                       Compare FROM column
PWCMPEND DS    H                       COMPARE TO column
PWWIPDSN DS    CL44                    WIP Dataset Name
*--------------------------------------------------------------------*
* Root information                                                   *
*--------------------------------------------------------------------*
PWROODSN DS    CL44                    Root Dataset Name
PWROOENV DS    CL8                     Root Environment
PWROOSYS DS    CL8                     Root System
PWROOSUB DS    CL8                     Root Subsystem
PWROOTYP DS    CL8                     Root Type
PWROOSTX DS    CL1                     Root Stage (ID or Number)
PWROOST# DS    CL1                     Root Stage Number (Y/N)
*--------------------------------------------------------------------*
* Derivation 1 information                                           *
*--------------------------------------------------------------------*
PWDV1DSN DS    CL44                    DER 1 Dataset Name
PWDV1ENV DS    CL8                     DER 1 Environment
PWDV1SYS DS    CL8                     DER 1 System
PWDV1SUB DS    CL8                     DER 1 Subsystem
PWDV1TYP DS    CL8                     DER 1 Type
PWDV1STX DS    CL1                     DER 1 Stage (ID or Number)
PWDV1ST# DS    CL1                     DER 1 Stage Number (Y/N)
*--------------------------------------------------------------------*
* Derivation 2 information                                           *
*--------------------------------------------------------------------*
PWDV2DSN DS    CL44                    DER 2 Dataset Name
PWDV2ENV DS    CL8                     DER 2 Environment
PWDV2SYS DS    CL8                     DER 2 System
PWDV2SUB DS    CL8                     DER 2 Subsystem
PWDV2TYP DS    CL8                     DER 2 Type
PWDV2STX  DS   CL1                     DER 2 Stage (ID or Number)
PWDV2ST#  DS   CL1                     DER 2 Stage Number (Y/N)
```

```
*-------------------------------------------------------------------*
* Other information                                                 *
*-------------------------------------------------------------------*
PWWIPMEM DS    CL10                    WIP Member name
PWROOMEM DS    CL10                    Root Member/Element name
PWROOLVL DS    H                       Root Level (if ENDEVOR)
PWROOVER DS    H                       Root Version (if ENDEVOR)
PWDV1MEM DS    CL10                    DER 1 Member/Element name
PWDV1LVL DS    H                       DER 1 Level (if ENDEVOR)
PWDV1VER DS    H                       DER 1 Version (if ENDEVOR)
PWDV2MEM DS    CL10                    DER 2 Member/Element name
PWDV2LVL DS    H                       DER 2 Level (if ENDEVOR)
PWDV2VER DS    H                       DER 2 Version (if ENDEVOR)
PWRUNTME DS    F                       Time of BUILD WIP action
PWRUNDTE DS    F                       Date of BUILD WIP action
PWPGENUM DS    PL2                     Report page number
PWRETCDE DS    F                       BUILD WIP return code
         SPACE ,
*-------------------------------------------------------------------*
* General statistics                                                *
*-------------------------------------------------------------------*
PWTOTINS DS    F                       Total Number of inserts
PWTOTDEL DS    F                       Total Number of deletes
PWTOTREC DS    F                       Total Number of WIP records
PWTOTCNF DS    F                       Total Number of conflicts
PWTOTCTA DS    F                       Total Number of Contention Areas
PWTOTC1R DS    F                       Total D1 records in conflict
PWTOTC2R DS    F                       Total D2 records in conflict
PWCOMPFC DS    H                       Complexity Factor for member
PWCOMP0  EQU   0                        - Complexity Factor 0
PWCOMP1  EQU   1                        - Complexity Factor 1
PWCOMP2  EQU   2                        - Complexity Factor 2
PWCOMP3  EQU   3                        - Complexity Factor 3
PWCOMP4  EQU   4                        - Complexity Factor 4
PWCOMP5  EQU   5                        - Complexity Factor 5
PWRSV1   DS    H                       Reserved
         SPACE ,
*-------------------------------------------------------------------*
* Common to both derivations                                        *
*-------------------------------------------------------------------*
PWTOT12I DS    F                       Total common insert recs (D1,2)
PWTOT12D DS    F                       Total common delete recs (D1,2)
PWTOT12C DS    F                       Total common changes (D1,D2)
PWTDCIBL DS    F                       Com - Total Number of ins blocks
PWTDCDBL DS    F                       Com - Total Number of del blocks
PWNDCISZ DS    F                       Com - Min size of insert blocks
PWXDCISZ DS    F                       Com - Max size of insert blocks
PWMDCISZ DS    F                       Com - Mean size of insert blocks
PWNDCDSZ DS    F                       Com - Min size of delete blocks
PWXDCDSZ DS    F                       Com - Max size of delete blocks
PWMDCDSZ DS    F                       Com - Mean size of delete blocks
         SPACE ,
```

```
*-------------------------------------------------------------------*
* Derivation 1 statistics                                           *
*-------------------------------------------------------------------*
PWTD1INS DS    F                      D1 - Total Number of inserts
PWTD1DEL DS    F                      D1 - Total Number of deletes
PWTD1IBL DS    F                      D1 - Total Number of ins blocks
PWTD1DBL DS    F                      D1 - Total Number of del blocks
PWTD1CHG DS    F                      D1 - Total Number of changes
PWND1ISZ DS    F                      D1 - Min size of insert blocks
PWXD1ISZ DS    F                      D1 - Max size of insert blocks
PWMD1ISZ DS    F                      D1 - Mean size of insert blocks
PWND1DSZ DS    F                      D1 - Min size of delete blocks
PWXD1DSZ DS    F                      D1 - Max size of delete blocks
PWMD1DSZ DS    F                      D1 - Mean size of delete blocks
         SPACE ,
*-------------------------------------------------------------------*
* Derivation 2 statistics                                           *
*-------------------------------------------------------------------*
PWTD2INS DS    F                      D2 - Total Number of inserts
PWTD2DEL DS    F                      D2 - Total Number of deletes
PWTD2IBL DS    F                      D2 - Total Number of ins blocks
PWTD2DBL DS    F                      D2 - Total Number of del blocks
PWTD2CHG DS    F                      D2 - Total Number of changes
PWND2ISZ DS    F                      D2 - Min size of insert blocks
PWXD2ISZ DS    F                      D2 - Max size of insert blocks
PWMD2ISZ DS    F                      D2 - Mean size of insert blocks
PWND2DSZ DS    F                      D2 - Min size of delete blocks
PWXD2DSZ DS    F                      D2 - Max size of delete blocks
PWMD2DSZ DS    F                      D2 - Mean size of delete blocks
*-------------------------------------------------------------------*
* Auto-Merge statistics                                             *
*-------------------------------------------------------------------*
PWAMRGRC DS    F                      Auto-Merge Return Code
PWAMRGFL DS    CL1                    Auto-Merge Indicator (Y/N)
$PBWDSLN EQU   *-$PBWSTAT
         MEND
```

D.2.1 Data Displayed

$PBWSTAT presents data in five sections.

- General source location information
- General statistic totals
- Data common to both derivations
- Derivation 1 statistics
- Derivation 2 statistics

Each section, and each field within that section, is explained below.

D.2.2 General Source Location Information

The first section of this control block provides general information about the WIP, Root, Derivation 1, and Derivation 2 source.

Field	Description
$PBWLEN	Size of the control block.
$PBWEYE	Eye-catcher, to help visually identify this control block.
$PBWVERS	Version to identify the control block.
PWTYPE	Type indicator for the control block. A type of **1** indicates that this is the **WIP Detail Statistics** control block.
PWREPL	Indicates your selection for the replace/no replace option for this member set. Valid values are: ■ Y — Yes ■ N — No
PWPRCMDE	Indicates the processing mode for this member set: ■ F — Foreground ■ B — Batch
PWCMPBEG	Indicates the column at which PDM should begin to compare Root, Derivation 1, and Derivation 2 information for the Build WIP process.
PWCMPEND	Indicates the last column of information that should be compared in the Build WIP process.
PWWIPDSN	The WIP data set name.

Note: The information presented in the next three groups of data (for the Root, Derivation 1, and Derivation 2) depends on whether you have a data set source location or an Endevor source location. If you use a data set name, the Endevor location fields are blank. If you use a Endevor location, the data set name field is blank.

Field	Description
PWROODSN	The Root data set name.
PWROOENV	The Root environment name.
PWROOSYS	The Root system name.
PWROOSUB	The Root subsystem name.
PWROOTYP	The Root element type.

Field	Description
PWROOST#	Indicates whether the value in the field above is an ID or a number: ■ Y — PWROOSTX is a stage number. ■ N — PWROOSTX is a stage ID.
PWROOSTX	The Root stage ID or number, depending on the value entered for this member set.
PWDV1DSN	The Derivation 1 data set name.
PWDV1ENV	The Derivation 1 environment name.
PWDV1SYS	The Derivation 1 system name.
PWDV1SUB	The Derivation 1 subsystem name.
PWDV1TYP	The Derivation 1 element type
PWDV1STX	The Derivation 1 stage ID or number, depending on the value entered for this member set.
PWDV1ST#	Indicates whether the value in the field above is an ID or a number: ■ Y — PWDV1STX is a stage number. ■ N — PWDV1STX is a stage ID.
PWDV2DSN	The Derivation 2 data set name.
PWDV2ENV	The Derivation 2 environment name.
PWDV2SYS	The Derivation 2 system name.
PWDV2SUB	The Derivation 2 subsystem name.
PWDV2TYP	The Derivation 2 element type.
PWDV2STX	The Derivation 2 stage ID or number, depending on the value entered for this member set.
PWDV2ST#	Indicates whether the value in the field above is an ID or a number: ■ Y — PWDV2STX is a stage number. ■ N — PWDV2STX is a stage ID.
PWWIPMEM	The WIP member name.
PWROOMEM	The Root member or element name.
PWROOLVL	The level of the Root source, if the Root uses an Endevor location
PWROOVER	The version of the Root source, if the Root uses an Endevor location.
PWDV1MEM	The Derivation 1 member or element name.
PWDV1LVL	The level of the Derivation 1 source, if Derivation 1 uses an Endevor location.

Field	Description
PWDV1VER	The version of the Derivation 1 source, if Derivation 1 uses a Endevor location.
PWDV2MEM	The Derivation 2 member or element name.
PWDV2LVL	The level of the Derivation 2 source, if Derivation 2 uses an Endevor location.
PWDV2VER	The version of the Derivation 2 source, if Derivation 2 uses a Endevor location.
PWRUNTME	The time at which the Build WIP process was executed.
PWRUNDTE	The date on which the Build WIP process was executed.
PWPGENUM	The page on which detail information for this member set appears in the PDM Build WIP Detail report.
PWRETCDE	The return code resulting from the Build WIP processing of this member set.

D.2.3 General Statistics Total

The second section of the control block provides a general summary for the entire WIP member set.

Field	Description
PWTOTINS	The total number of lines marked as insertions to the Root program, from both Derivation 1 and Derivation 2.
PWTOTDEL	The total number of lines marked as deletions to the Root program, from both Derivation 1 and Derivation 2.
PWTOTREC	The total number of records in this WIP member set.
PWTOTCNF	The total number of conflict areas in this WIP member. Refer to Chapter 1, “An Introduction to PDM” for a definition of a conflict area.
PWTOTCTA	The total number of contention areas in this WIP member. Refer to Chapter 1, “An Introduction to PDM” for a definition of a contention area.
PWTOTC1R	The total number of Derivation 1 records involved in conflicts in this member set.
PWTOTC2R	The total number of Derivation 2 records involved in conflicts in this member set.

Field	Description
PWCOMPFC	The complexity factor of this WIP member: *Value* *Label* *Complexity Factor* 0 PWCOMP0 0 1 PWCOMP1 1 2 PWCOMP2 2 3 PWCOMP3 3 4 PWCOMP4 4 5 PWCOMP5 5 Refer to Chapter 1, "An Introduction to PDM" for a definition of complexity factor.

D.2.4 Data Common to Both Derivations

This section of the control block provides information for data that is common to both derivations.

Field	Description
PWTOT12I	Indicates the number of lines marked as insertions, from both derivations, at the same location with respect to the Root program (that is, insertions in common).
PWTOT12D	Indicates the number of lines marked as deletions, from both derivations, at the same location with respect to the Root program (that is, deletions in common).
PWTOT12C	Indicates the number of changes made to the same records of the Root program, by both derivations. This value refers to modifications made to an existing record (such as a change in field size or field name), rather than an insertion or a deletion of a record.
PWTDCIBL	Indicates the total number of blocks of insertion lines, made by both derivations, in common (that is, insertions at the same location, with respect to the Root program). Refer to 6.3, "PDM Build WIP Detail Report" on page 6-4 for the definition of block.
PWTDCDBL	Indicates the total number of blocks of deletion lines, made by both derivations, in common (that is, deletions at the same location, with respect to the Root program). Refer to 6.3, "PDM Build WIP Detail Report" on page 6-4 for the definition of block.
PWNDCISZ	The minimum size (smallest) of a common insertion block in this WIP member.
PWXDCISZ	The maximum size (largest) of a common insertion block in this WIP member.

Field	Description
PWMDCISZ	The mean average size of a common insertion block in this WIP member.
PWNDCDSZ	The minimum size (smallest) of a common deletion block in this WIP member.
PWXDCDSZ	The maximum size (largest) of a common deletion block in this WIP member.
PWMDCDSZ	The mean average size of a common deletion block in this WIP member.

D.2.5 Derivation 1 Statistics

This section of the control block presents information pertaining to Derivation 1 only.

Field	Description
PWTD1INS	Indicates the total number of lines marked as insertions to the Root program by Derivation 1.
PWTD1DEL	Indicates the total number of lines marked as deletions to the Root program by Derivation 1.
PWTD1IBL	Indicates the number of blocks of insertion lines from Derivation 1.
PWTD1DBL	Indicates the number of blocks of deletion lines from Derivation 1.
PWTD1CHG	Indicates the number of changes made to existing records in the Root program by Derivation 1.
PWND1ISZ	Indicates the minimum size (smallest) of an insertion block from Derivation 1.
PWXD1ISZ	Indicates the maximum size (largest) of an insertion block from Derivation 1.
PWMD1ISZ	Indicates the mean average size of an insertion block from Derivation 1.
PWND1DSZ	Indicates the minimum size (smallest) of a deletion block from Derivation 1.
PWXD1DSZ	Indicates the maximum size (largest) of a deletion block from Derivation 1.
PWMD1DSZ	Indicates the mean average size of a deletion block from Derivation 1.

D.2.6 Derivation 2 Statistics

This section of the control block presents information pertaining to Derivation 2 only.

Field	Description
PWTD2INS	Indicates the total number of lines marked as insertions to the Root program by Derivation 2.
PWTD2DEL	Indicates the total number of lines marked as deletions to the Root program by Derivation 2.
PWTD2IBL	Indicates the number of blocks of insertion lines from Derivation 2.
PWTD2DBL	Indicates the number of blocks of deletion lines from Derivation 2.
PWTD2CHG	Indicates the number of changes made to existing records in the Root program by Derivation 2.
PWND2ISZ	Indicates the minimum size (smallest) of an insertion block from Derivation 2.
PWXD2ISZ	Indicates the maximum size (largest) of an insertion block from Derivation 2.
PWMD2ISZ	Indicates the mean average size of an insertion block from Derivation 2.
PWND2DSZ	Indicates the minimum size (smallest) of a deletion block from Derivation 2.
PWXD2DSZ	Indicates the maximum size (largest) of a deletion block from Derivation 2.
PWMD2DSZ	Indicates the mean average size of a deletion block from Derivation 2.

D.2.7 Auto-Merge information

This section of the control block presents information pertaining WIP members that were automatically merged.

Field	Description
PWAMRGRC	The merge return code.
PWAMRGFL	The auto-merge indicator. It is set to **Y** if the WIP member was automatically merged.

D.3 $PBWSSUM: WIP Summary Statistics

The $PBWSSUM control block presents summary information for the entire WIP data set, and reflects much of the statistical information provided on the PDM Build WIP Summary report. Only one $PBWSSUM control block is written for each Build action.

```
*--------------------------------------------------------------------*
*                                                                    *
* Macro: $PBWSSUM                                                    *
*                                                                    *
* Function: The $PBWSSUM macro maps the PDM BUILD WIP Summary stat-*
*  istics record.                                                    *
*                                                                    *
*--------------------------------------------------------------------*
$PBWSSUM DSECT
$PWSLEN  DC    Y($PWSDSLN)             Length of the structure
$PWSEYE  DC    CL4'SSUM'               Eye catcher
$PWSVERS DC    Y(1)                    Version number
WSTYPE   DC    Y(2)                    Record Type: WIP Summary
WSWIPDSN DS    CL44                    WIP dataset name
WSMBRSEL DS    F'0'                    Members selected for processing
WSMBRSUC DS    F'0'                    Members successfully processed
WSMBRREP DS    F'0'                    Members with REPLACE conflicts
WSMBRERR DS    F'0'                    Members in error
WSMNOICD DS    F'0'                    Members with no Inserts/deletes
WSMNOCON DS    F'0'                    Members with no conflicts
WSMCOMP5 DS    F'0'                    Members, complexity factor 5
WSMCOMP4 DS    F'0'                    Members, complexity factor 4
WSMCOMP3 DS    F'0'                    Members, complexity factor 3
WSMCOMP2 DS    F'0'                    Members, complexity factor 2
WSMCOMP1 DS    F'0'                    Members, complexity factor 1
WSMCOMP0 DS    F'0'                    Members, complexity factor 0
WSMWPMRG DS    F'0'                    Members Auto-Merged
WSHWIPRC DS    F'0'                    Highest BUILD return code
$PWSDSLN EQU   *-$PBWSSUM
         MEND
```

Each field in the control block is described below.

Field	Description
$PWSLEN	Size of the control block.
$PWSEYE	Eye-catcher, to help visually identify this control block.
$PWSVERS	Version to identify the control block.
WSTYPE	Type indicator for the control block. A type of **2** indicates that this is the **WIP Summary** control block.
WSWIPDSN	The WIP data set name.

Field	Description
WSMBRSEL	Indicates the total number of members selected for processing.
WSMBRSUC	Indicates the total number of members processed successfully (with a return code of 0000).
WSMBRREP	Indicates the number of members flagged as "replace conflicts;" that is, like-named members exist in the WIP data set, but the replace/no replace option has been set to "no replace." These members are not processed.
WSMBRERR	Indicates the number of members flagged as "in error." These members, if processed, should be considered invalid.
WSMNOICD	Indicates the number of members with no insertions or deletions from either derivation (that is, no changes at all were made to the member).
WSMNOCON	Indicates the number of members to which changes were made, but with no conflicts resulting from those modifications.
WSMCOMP5	Indicates the number of WIP members with a complexity factor of 5.
WSMCOMP4	Indicates the number of WIP members with a complexity factor of 4.
WSMCOMP3	Indicates the number of WIP members with a complexity factor of 3.
WSMCOMP2	Indicates the number of WIP members with a complexity factor of 2.
WSMCOMP1	Indicates the number of WIP members with a complexity factor of 1.
WSMCOMP0	Indicates the number of WIP members with a complexity factor of 0.
WSMWPMRG	Indicates the number of WIP members that were automatically merged.
WSHWIPRC	The highest build return code.

D.4 $PBMSTAT: Merge Detail Statistics

The $PBMSTAT control block presents detail information about each member set involved in the Merge process, and reflects much of the statistical information provided in the PDM Merge Output Detail Report. One $PBMSTAT control block is written for each member set.

```
*------------------------------------------------------------------------*
*                                                                        *
* Macro: $PBMSTAT                                                        *
*                                                                        *
* Function: The $PBMSTAT macro is used to map the PDM Merge Output *
*  statistics record.  An Output statistics record is created for     *
*  each WIP member processed by the PDM MERGE OUTPUT action.          *
*                                                                        *
*------------------------------------------------------------------------*
$PBMSTAT DSECT
$PBMLEN  DC    Y($PBMDSLN)             Length of the structure
$PBMEYE  DC    CL4'STAT'               Structure Identifier
$PBMVERS DC    Y(1)                    Structure Version number
PMTYPE   DC    Y(3)                    Structure Type identifier
PMREPL   DS    CL1                     Global Replace option (Y/N)
PMPRCMDE DS    CL1                     Processing Mode (F/B)
PMWIPDSN DS    CL44                    WIP Dataset Name
PMOUTDSN DS    CL44                    Merge Output Dataset name
PMWIPMEM DS    CL10                    WIP Member name
PMOUTMEM DS    CL10                    Merge Output Member name
PMRUNTME DS    F                       Time of Merge operation
PMRUNDTE DS    F                       Date of Merge operation
         SPACE ,
*------------------------------------------------------------------------*
* Merge Output General Statistics                                        *
*------------------------------------------------------------------------*
PMPGENUM DS    PL2                     Report Page Number
PMWIPMRG DS    CL1                     If 'Y', Merge was run as part  X
                                        of Build WIP processing
PMWIPDEL DS    CL1                     If 'Y', the WIP member was del-X
                                        eted by the Merge operation
PMRETCDE DS    F                       Overall return code for member
PMTWPREC DS    F                       Total number of WIP records
PMTMGREC DS    F                       Total number of Merged records
PMTERREC DS    F                       Total number of invalid records
$PBMDSLN EQU   *-$PBMSTAT
         MEND
```

D.4.1 Data Display

$PBMSTAT presents data in two sections:

- General source location information
- General statistics

Each section, and each field within that section, is explained below.

D.4.2 General Source Location Information

The first section of this control block provides general information about each WIP and output source member.

Field	Description
$PBMLEN	Size of the control block.
$PBMEYE	Eye-catcher for this control block.
$PBMVERS	Version to identify the control block.
PMTYPE	Type indicator for the control block. A type of **3** indicates that this is the Merge Detail Statistics control block.
PMREPL	Indicates your selection for the replace/no replace option for this member set: ■ Y — Yes ■ N — No
PMPRCMDE	Indicates the processing mode for this member set: ■ F — Foreground ■ B — Batch
PMWIPDSN	The WIP data set name.
PMOUTDSN	The output file data set name.
PMWIPMEM	The WIP member name.
PMOUTMEM	The output file member name.
PMRUNTME	The time at which the Merge process was executed.
PMRUNDTE	The date on which the Merge process was executed.

D.4.3 General Statistics

The second section of this control block provides statistical data for each member involved in the Merge process.

Field	Description
PMPGENUM	The page on which detail information for this member appears in the PDM Merge Output Detail Report.
PMWIPMRG	Set to **Y** if the merge operation was done as part of Build WIP processing.
PMWIPDEL	Set to **Y** if the WIP member was deleted by the merge operation.
PMRETCDE	The return code resulting from the Merge processing of this member.
PMTWPREC	The total number of records contained in this WIP member.

Field	Description
PMTMGREC	The total number of records contained in the output source file (merged file).
PMTERREC	The total number of invalid WIP records detected in this particular WIP member.

D.5 $PBMSSUM: Merge Summary Statistics

The $PBMSSUM control block presents summary information for the WIP and output data sets involved in the Merge process. Only one $PBMSSUM control block is written for each Merge action.

```
*------------------------------------------------------------------------*
*                                                                        *
* Macro: $PBMSSUM                                                        *
*                                                                        *
* Function: The $PBMSSUM macro maps the PDM Merge Summary statistics     *
*  record.  The Summary statistics record contains summary inform-       *
*  ation about the MERGE OUTPUT operation.                               *
*                                                                        *
*------------------------------------------------------------------------*
$PBMSSUM DSECT
$PMSLEN  DC    Y($PMSDSLN)             Length of the structure
$PMSEYE  DC    CL4'SSUM'               Structure identifier
$PMSVERS DC    Y(1)                    Structure version number
MSTYPE   DC    Y(4)                    Structure type identifier
MSOUTDSN DS    CL44                    Merge Output dataset name
MSMBRSEL DS    F'0'                    Members selected for processing
MSMBRSUC DS    F'0'                    Members successfully processed
MSMBRREP DS    F'0'                    Members with replace conflicts
MSMBRERR DS    F'0'                    Members in error
MSWIPDEL DS    F'0'                    Number of WIP members deleted
$PMSDSLN EQU   *-$PBMSSUM
         MEND
```

Each field in the control block is described below:

Field	Description
$PMSLEN	Size of the control block.
$PMSEYE	Eye-catcher for this control block.
$PMSVERS	Version to identify the control block.
MSTYPE	Type indicator for the control block. A type of **4** indicates that this is the Merge Summary control block.
MSOUTDSN	The Merge Output data set name.
MSMBRSEL	Indicates the total number of members selected for processing.
MSMBRSUC	Indicates the total number of members processed successfully (with a return code of 0000).
MSMBRRE	Indicates the number of members flagged as "replace conflicts;" that is, like-named members exist in the WIP data set, but the replace/no replace option has been set to "no replace." These members are not processed.

Field	Description
MSMBRERR	Indicates the number of members flagged as "in error." These members, if processed, should be considered invalid.
MSWIPDEL	Indicates the number of WIP members deleted by merge processing.

Index

Special Characters

$PBMSSUM control block (Merge Summary Statistics) D-17
$PBMSTAT control block (Merge Detail Statistics) D-14
$PBWSSUM control block (WIP Summary Statistics) D-12
$PBWSTAT control block (WIP Detail Statistics) D-3

A

Accessing PDM 1-11
AllFusion Endevor Change Manager Parallel Development Option menu 1-11, 1-12, 2-2, 3-5, 4-5, 5-3, 5-6, 5-7, 5-8
Auto-Merge Information D-11

B

Batch PDM
  generating build WIP and merge requests 5-2
  generating build WIP requests option 1 5-6
  generating merge requests option 3 5-7
  specifying a request data set 5-3
  submitting batch requests for execution option 4 5-8
Batch requests for execution option 4, submitting 5-8
BC1GJCL2 JCL sample A-3
Beginning build of WIP information 6-5
Beginning merge of WIP member information 6-15
Browsing request data set 5-9
BUILD MASK syntax statement B-10
BUILD PDM syntax statement B-5
Build WIP
  allocating a WIP library 2-6
  creating a merge output file 2-19
  creating WIP files or SCL 2-31
  excluding a WIP member 2-23
  including matrix rows 2-24
  options 2-5
Build WIP *(continued)*
  positioning the matrix 2-30
  printing the matrix 2-29
  process 2-2
  renaming a WIP member 2-25
  selecting an element 2-12
  sorting the matrix 2-31
  specifying a derivation 1 library or Endevor location 2-14
  specifying a derivation 1 member or element 2-14
  specifying a derivation 2 library or Endevor location 2-17
  specifying a derivation 2 member or element 2-17
  specifying a root member or element 2-11
  specifying a WIP library and build WIP options 2-4
  specifying a WIP member 2-8
  specifying derivation 1 2-14
  specifying derivation 2 2-17
  specifying the root 2-9
  using the WIP member selection matrix 2-21
Build WIP - Member Selection list 2-11
Build WIP requests option 1, generating 5-6
Build WIP requests, generating 5-2
BUILD WIP summary information 6-10

C

C1MSGS1 DD statement 6-2
C1MSGS2 DD statement 6-2
Commands
  CREATE 2-31
  EXCLUDE 2-23
  INCLUDE 2-24
  LOCATE 2-30
  PRTMAT 2-29
  RENAME 2-25
  SORT 2-31
  WIP edit 3-7

Common deletion 1-9
Common insertion 1-9
Complex parallel development 1-4
Complexity factor 1-9
Concurrently developed software, integrating 1-5
Conflict 1-9
Conflict area 1-9
Conflicts information 6-7
Contention area 1-9
CREATE command 2-31
Creating
  merge output file 2-19, 4-5
  WIP files or SCL 2-31

D

Derivation 1 file name 1-9
Derivation 1 library or Endevor location, specifying 2-14
Derivation 1 member or element, specifying 2-14
Derivation 1 or derivation 2 member, moving 2-26
Derivation 1 Specification Panel 2-14
Derivation 1, specifying 2-14
Derivation 2 file name 1-9
Derivation 2 library or Endevor location, specifying 2-17
Derivation 2 member or element, specifying 2-17
Derivation 2 Specification Panel 2-17
Derivation 2, specifying 2-17
Detail statistics 6-5, 6-15
Disable the build WIP and merge performance enhancement option 1-13
Displaying
  beginning build of WIP information 6-5
  beginning merge of WIP member information 6-15
  BUILD WIP summary information 6-10
  conflicts information 6-7
  detail statistics 6-5, 6-15
  highest BUILD return code 6-11
  highest MERGE return code 6-17
  inserts and deletes information 6-6
  merge summary information 6-17
  output data set information 6-16
  relative change information 6-5
  return code information 6-5, 6-15
  WIP data set library information 6-9, 6-16
  WIP data set member information 6-10
  WIP member information 6-16

E

Edit - Entry Panel 3-5, 3-14
Edit Macronames listed with member names C-3
Edit macros C-3
  extending Edit Macros C-4
Editing
  request data set 5-10
  WIP file 3-5
Element Selection List 2-12
EXCLUDE command 2-23

G

Generating
  Build WIP and merge requests 5-2
  build WIP requests option 1 5-6
  merge requests option 3 5-7

H

Highest BUILD return code 6-11
Highest MERGE return code 6-17

I

INCLUDE command 2-24
Inserts and deletes information 6-6
Integrating concurrently developed software 1-5

J

JCL samples
  BC1GJCL2 A-3

L

LOCATE command 2-30

M

Macronames listed with member names C-3
Matrix
  positioning 2-30
  printing 2-29
  sorting 2-31
Matrix rows, including 2-24
Merge Detail Statistics (control block $PBMSTAT) D-14
MERGE MASK PDM syntax statement B-15
Merge output file
  creating 2-19, 4-5
  description 4-2

Merge output file *(continued)*
example 4-4
file name 1-9
Merge Output Member Selection List panel 4-9, 4-10, 4-11
Merge Output Specification Panel 2-19, 4-6, 4-8
MERGE PDM syntax statement B-12
Merge requests option 3, generating 5-7
Merge requests, generating 5-2
Merge summary information 6-17
Merge Summary Statistics (control block $PBMSSUM) D-17
Merge Work In Process File panel 4-5, 4-7
Merging a WIP file 4-5
Moving a derivation 1 or derivation 2 member 2-26

O

Output data set information 6-16

P

Panels
AllFusion Endevor Change Manager Parallel Development Option menu 1-11, 1-12, 2-2, 3-5, 4-5, 5-3, 5-6, 5-7, 5-8
Build WIP - Member Selection List 2-11
Derivation 1 Specification Panel 2-14
Derivation 2 Specification Panel 2-17
Edit - Entry Panel 3-5, 3-14
Element Selection List 2-12
Merge Output Member Selection List 4-9, 4-10, 4-11
Merge Output Specification Panel 2-19, 4-6, 4-8
Merge Work In Process File 4-5, 4-7
Parallel Development Option WIP Summary 3-2
PDM Batch Options Menu 5-8, 5-11
PDM User Defaults 1-12
Request Dataset Allocation Panel 5-4
Root Specification Panel 2-9
Summary of Levels 2-33, 2-34
WIP Dataset Allocation Panel 2-6
WIP Member Selection Matrix 1-10, 2-21, 2-23, 2-24, 2-25, 2-26, 2-27, 2-31, 2-34
WIP Specification Panel 2-4
Parallel development
complex 1-4
description 1-3
simple 1-3
Parallel Development Option WIP Summary 3-2
Parallel Development Option. See PDM
PDM
accessing 1-11
file names 1-9
how it works 1-7
specifying user defaults 1-12
terms 1-9
tools for project analysis 1-6
PDM Batch Options Menu 5-8, 5-11
PDM Build WIP Detail report 6-4
PDM Build WIP Summary report 6-9
PDM Merge Output Detail report 6-14
PDM Merge Output Summary report 6-16
PDM Syntax Request Report for Build WIP 6-3
PDM Syntax Request Report for Merge 6-13
PDM syntax statements
BUILD B-5
BUILD MASK B-10
description B-3
MERGE B-12
MERGE MASK B-15
STATS OFF B-16
structure of B-3
PDM User Defaults panel 1-12
PRTMAT command 2-29
PRTMAT output defaults 1-13

R

Relative change information 6-5
RENAME command 2-25
Reports
PDM Build WIP Detail 6-4
PDM Build WIP Summary 6-9
PDM Merge Output Detail 6-14
PDM Merge Output Summary 6-16
PDM Syntax Request Report for Build WIP 6-3
PDM Syntax Request Report for Merge 6-13
Request data set
browsing 5-9
editing 5-10
specifying 5-3
submitting for batch processing 5-11
Request Dataset Allocation Panel 5-4
Return code information 6-5, 6-15
Root file name 1-9
Root member or element, specifying 2-11
Root Specification Panel 2-9
Root, specifying 2-9

S

Saving WIP member after editing 3-14
SCL, creating 2-31
Simple parallel development 1-3
SORT command 2-31
Specifying
  derivation 1 2-14
  derivation 1 library or Endevor location 2-14
  derivation 1 member or element 2-14
  derivation 2 2-17
  derivation 2 library or Endevor location 2-17
  derivation 2 member or element 2-17
  request data set 5-3
  root 2-9
  root member or element 2-11
  user defaults 1-12
  WIP library and build WIP options 2-4
  WIP member 2-8
Statistics file, file name 1-9
STATS OFF PDM syntax statement B-16
Submitting
  batch requests for execution option 4 5-8
  request data set for batch processing 5-11
Summary of Levels panel 2-33, 2-34
Syntax validation 6-3

U

Updates, vendor application 1-4
User defaults for PDM, specifying 1-12

V

Vendor application updates 1-4

W

WIP data set library information 6-9, 6-16
WIP data set member information 6-10
WIP Dataset Allocation Panel 2-6
WIP Detail Statistics (control block $PBWSTAT) D-3
WIP edit commands
  WIPCHANG 3-7, 3-11
  WIPCON 3-7, 3-12
  WIPCOUNT 3-7, 3-9
  WIPHELP 3-7
  WIPLDEL 3-7, 3-9
  WIPMERGE 3-7, 3-12
  WIPPARA 3-7, 3-11
  WIPSHOW 3-7, 3-10
  WIPUNDEL 3-7, 3-9
WIP file
  annotations 3-3
  creating 2-31
  editing 3-5
  example 3-3
  format 3-2
  header 3-2
  merging 4-5
WIP library
  allocating 2-6
  specifying 2-4
WIP member
  excluding 2-23
  renaming 2-25
  saving after editing 3-14
  specifying 2-8
WIP member information 6-16
WIP Member Selection Matrix 2-21, 2-23, 2-24, 2-25, 2-26, 2-27, 2-31, 2-34
WIP Member Selection Matrix panel 1-10
WIP Specification Panel 2-4
WIP Summary Statistics (control block $PBWSSUM) D-12
Work in process (WIP) file, file name 1-9